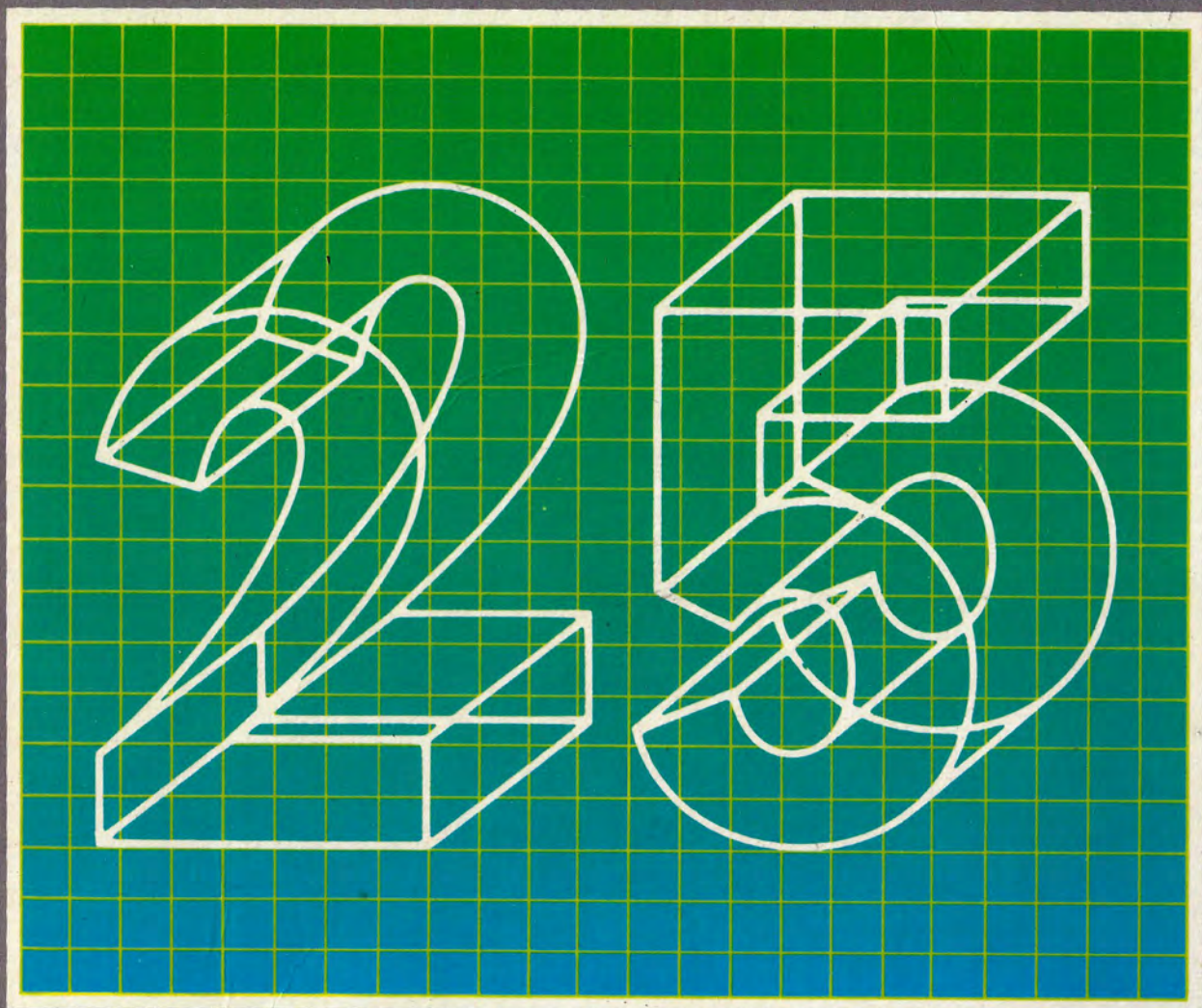


BIBLIOTECA PRACTICA

TALLER DE INFORMATICA

TRUCOS - SPRITES
BRICOLAGE DEL HARD
AULA ABIERTA
LOGO - TERMINOLOGIA

PROGRAMAS VALIDOS PARA AMSTRAD,
IBM, SPECTRUM, COMMODORE Y MSX



DIVERTIDOS PROGRAMAS

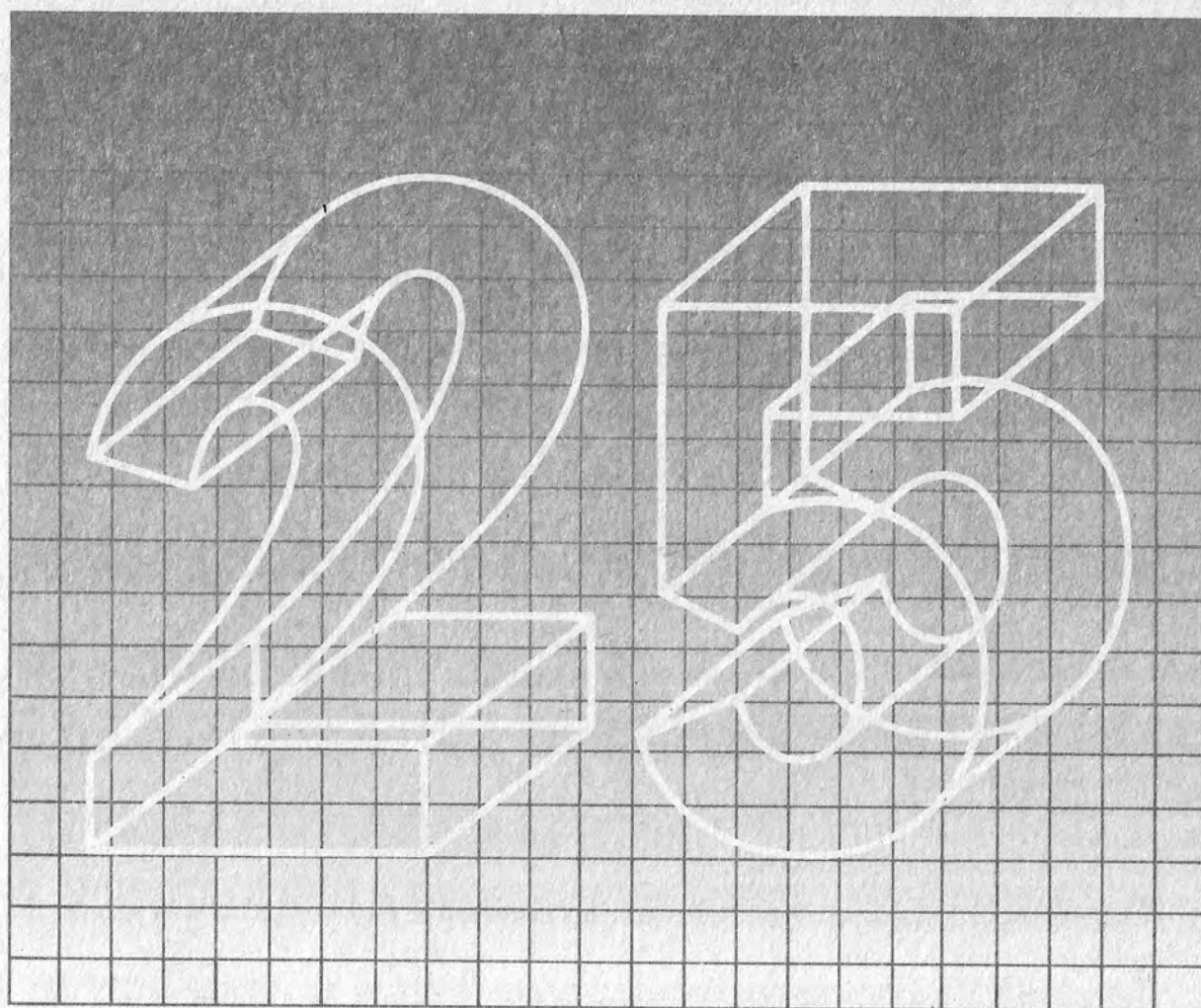
DE BOTANICA, LITERATURA, FISICA, ETC.

EDICIONES SIGLO CULTURAL

BIBLIOTECA PRACTICA TALLER DE INFORMATICA

TRUCOS - SPRITES
BRICOLAGE DEL HARD
AULA ABIERTA
LOGO - TERMINOLOGIA

PROGRAMAS VALIDOS PARA AMSTRAD,
IBM, SPECTRUM, COMMODORE Y MSX



EDICIONES SIGLO CULTURAL

Una publicación de

EDICIONES SIGLO CULTURAL, S. A.

Director-editor:

RICARDO ESPAÑOL CRESPO.

Gerente:

ANTONIO G. CUERPO.

Directora de producción:

MARIA LUISA SUAREZ PEREZ.

Director de la colección:

JOSE ARTECHE, Ingeniero de Telecomunicación

Diseño:

BRAVO-LOFISH.

Maquetación:

D. S. M.

Dibujos:

JOSE OCHOA

Tomo 25 «Experiencias y prácticas en logo», Carlos Albet, Técnico de Informática; Adoración Llena Jubelo, Técnica de Informática. «Manejo de sprites y elementos gráficos», «Trucos y rutinas básicas», Francisco Morales. Técnico de Informática. «Aprende con el ordenador», AULA DE INFORMATICA APLICADA: Angel Leira, Licenciado en Geografía e historia; Jorge Thomas, Técnico en Informática; Alejandro Marcos, Licenciado en Química. «El Taller del Hardware», Carlos Rey, Ingeniero Industrial. «Pequeña historia de la Informática», «Temas monográficos de vanguardia», «Terminología», «Vocabulario de Informática», Blanca Arbizu, Técnico de Informática.

Ediciones Siglo Cultural, S. A.

Dirección, redacción y administración:

Pedro Teixeira, 8-2.º. Teléf. 810 52 13. 28020 Madrid.

Publicidad:

Gofar Publicidad, S. A. Benito de Castro, 12 bis. 28028 Madrid.

Distribución en España:

COEDIS, S. A. Valencia, 245. Teléf. 215 70 97. 08007 Barcelona.

Delegación en Madrid: Serrano, 165. Teléf. 411 11 48.

Distribución en Ecuador: Muñoz Hnos.

Distribución en Perú: DISELPESA.

Distribución en Chile: Alfa Ltda.

Importador exclusivo Cono Sur:

CADE, S.R.L. Pasaje Sud América, 1532. Teléf.: 21 24 64.

Buenos Aires - 1.290. Argentina.

Todos los derechos reservados. Este libro no puede ser, en parte o totalmente, reproducido, memorizado en sistemas de archivo, o transmitido en cualquier forma o medio, electrónico, mecánico, fotocopia o cualquier otro, sin la previa autorización del editor.

ISBN del tomo: 84-7688-133-9

ISBN de la obra: 84-7688-047-2

Fotocomposición:

ARTECOMP, S. A. Albarracín, 50. 28037 Madrid.

Imprime:

MATEU CROMO. PINTO (Madrid).

© Ediciones Siglo Cultural, S. A., 1986

Depósito legal: M-43.185-1986

Printed in Spain - Impreso en España.

Suscripciones y números atrasados:

Ediciones Siglo Cultural, S. A.

Pedro Teixeira, 8-2.º. Teléf. 810 52 13. 28020 Madrid.

Julio, 1987

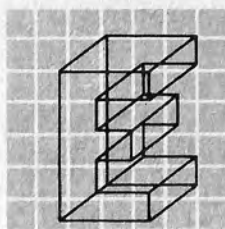
P.V.P. Canarias: 355.—Ptas.

INDICE

■ EXPERIENCIA Y PRACTICAS EN LOGO	5
■ MANEJO DE SPRITES Y ELEMENTOS GRAFICOS	16
■ TRUCOS Y RUTINAS BASICAS	24
■ EL TALLER DEL HARDWARE	32
■ APRENDER CON EL ORDENADOR	38
■ PEQUEÑA HISTORIA DE LA INFORMATICA	51
■ TEMAS MONOGRAFICOS DE VANGUARDIA	55
■ TERMINOLOGIA	59
■ VOCABULARIO DE INFORMATICA	60

EXPERIENCIA Y PRACTICAS EN LOGO

■ Escribir en la pantalla



N el tomo anterior conocimos una nueva primitiva Logo con la que podemos escribir en cualquier parte de la pantalla una o varias palabras. En esta ocasión, vamos a conocer algunas de las posibilidades que nos

ofrece esta orden y cómo actúa en situaciones determinadas.

La sintaxis de esta orden es:

PONCURSOR [C F]

C y F son dos números que indican la columna (C), y la fila (F) en donde se desea que un determinado texto se empiece a escribir.

Tanto el número de columna como el número de fila no debe sobrepasar el máximo de columnas y filas que tenga la versión del Logo con la que se esté trabajando.

Si se da un número mayor a este máximo, por ejemplo:

?PONCURSOR [100 100]

El Logo nos dará el siguiente mensaje de error:

PONCURSOR NO ACEPTA 100 COMO DATO.

Este mensaje nos avisa que se ha introducido un número de columna o fila superior al que posee.

Una situación que se nos puede presentar es la siguiente:

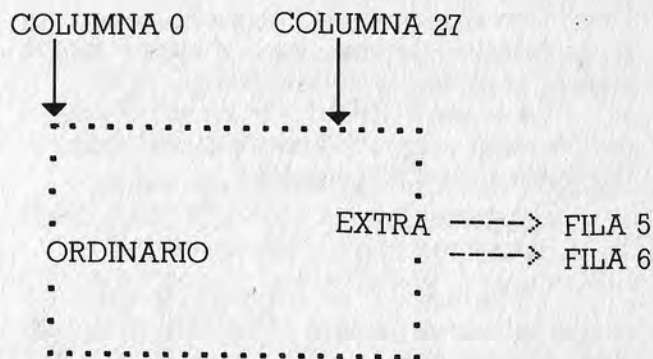
Supongamos que deseamos escribir la palabra EXTRAORDINARIO a partir de la co-

lumna número 27 y en la fila número 5. Para ello tenemos que introducir las siguientes órdenes:

?PONCURSOR [27 5]

?ESCRIBE "EXTRAORDINARIO

Si la versión de Logo con la que se está trabajando dispone de 32 columnas, obtendríamos lo siguiente:



La palabra comenzará a escribirse en la columna 27 en donde irá situado el carácter E. Al llegar a la columna 32 y escribirse el carácter A, la palabra se partirá, ya que no existen más columnas disponibles para seguir escribiendo.

El resto de los caracteres que forman la palabra se escribirían al principio de la siguiente fila.

Por tanto, si la longitud de la palabra a escribir es mayor al número de columnas que existen para poder escribirse, la palabra se partirá y la parte que exceda se escribirá al principio de la siguiente fila.

Supongamos ahora que introducimos las siguientes órdenes:

Podemos abrir una ventana de texto en la pantalla gráfica.

EXPERIENCIA Y PRACTICAS EN LOGO

```
?PONCURSOR [10 5]
?ESCRIBE "BUENA
?ESCRIBE "SUERTE
```

COLUMNA 0	CLUMNA 10	
↓	↓	
.....		
.	.	
.	.	
.	BUENA	----> FILA 5
SUERTE	.	----> FILA 6
.	.	
.	.	
.....		

La primera palabra se escribe a partir de posición indicada y la segunda se escribe en la siguiente fila y al principio.

Esto significa que la orden PONCURSOR sólo afecta a la primera orden ESCRIBE que hay a continuación.

Una vez que se ha escrito una palabra en la posición determinada, el cursor queda situado al principio de la siguiente línea.

La orden PONCURSOR no admite como dato el valor asignado a una determinada variable mediante la orden HAZ:

```
?HAZ "V 10
?PONCURSOR [:V 10]
```

Obtendremos un mensaje de error en el que se nos avisa que PONCURSOR no admite :V como dato.

Más adelante veremos la forma de poner datos variables a esta orden.

Una orden que está ligada a PONCURSOR es:

```
?CURSOR
```

Esta orden nos dirá la posición que tiene el cursor en ese momento.

Devuelve dos números. El primero corresponde a la columna y el segundo a la fila de la posición del cursor.

Si definimos:

```
?PARA EJEMPLO
>PONCURSOR [10 10]
>ESCRIBE CURSOR
>FIN
```

y ahora lo ejecutamos:

```
?EJEMPLO
```

en la pantalla aparecerán los números:

```
10 10
```

Si en el procedimiento añadimos la siguiente orden:

```
?PARA EJEMPLO
>PONCURSOR [10 10]
>ESCRIBE "HOLA
>ESCRIBE CURSOR
>FIN
```

En la pantalla aparecerá la palabra HOLA escrita a partir de la décima columna y en la décima fila, pero los números que indican la posición del cursor serán ahora:

```
0 11
```

Es decir, el cursor queda situado después de escribir la palabra en la columna cero y en la fila siguiente, la 11.

Otra relacionada con todo lo que estamos viendo es:

```
PONANCHURA
```

Esta orden, dependiendo de la versión de Logo con la que se trabaje, tiene una acción u otra:

```
—MSX
?PONANCHURA [F F]
```

Sitúa una ventana de texto en la pantalla gráfica.

Los dos números que se indican corresponden a la primera y a la última fila, respectivamente, donde se desea abrir la ventana de texto.

Estos dos valores pueden oscilar entre 0 y 23.

PC-COMPATIBLES

```
?PONANCHURA n
```

Determina la anchura que han de tener las filas en pantalla.

Entendemos por anchura de fila el número de caracteres que puede contener cada una.

En definitiva, esta orden determina el número de columnas que va a tener la pantalla.

Si una palabra no cabe en una fila, la parte que sobra se escribe en la fila siguiente.

El valor de n tiene que estar comprendido entre 2 y 80.

Inicialmente, el valor de n es de 40 para los monitores en color o TV y de 80 para los monitores monocromos.

SPECTRUM

Esta orden no existe para este ordenador.

Esta orden dispone de su correspondiente primitiva, que nos dice la anchura establecida.

?ANCHURA

Devuelve dos números (para los MSX), que corresponden a la primera y a la última fila de la ventana de texto abierta anteriormente.

Devuelve un número (para los PC-COMPATIBLES), que representa el número de columnas fijadas.

■ Gestión de archivos

A continuación detallamos una serie de primitivas que nos permitirán gestionar diversos tipos de archivos.

—ARCHIVO? "nombrearchivo (sólo admite disco)

Esta primitiva devuelve CIERTO si existe el archivo indicado en nombrearchivo, si no, devuelve FALSO.

Supón que en el disco activo existe un archivo con el nombre TEXTO.LOG.

?ESCRIBE ARCHIVO? "TEXTO.LOG
CIERTO

Efectivamente, el archivo TEXTO.LOG existe.

—CARGAB "nombrearchivo dirección

Carga en memoria el archivo "nombrearchivo" de tipo binario, a partir de la dirección especificada.

La dirección de carga debe ser un número comprendido entre 0 y 65535.

?CARGAB "ROUTINA.BIN 100

Cargamos en memoria a partir de la dirección 100, el archivo ROUTINA.BIN.

NOTA: En el LOGO del SPECTRUM la sintaxis de esta primitiva es:

.CARGAB "nombrearchivo dirección
—GUARDAB "nombrearchivo dirección1 dirección2

Guarda en el archivo "nombrearchivo" la parte de memoria comprendida entre dirección1 y dirección2.

?GUARDA "ROUTINA.BIN 100 200

Guardamos en el archivo ROUTINA.BIN lo que se halla en la memoria de la dirección 100 a 200.

NOTA: La sintaxis de esta primitiva en la versión para el SPECTRUM es:

GUARDAB "nombrearchivo dirección1
 n

Guarda en nombrearchivo los n octetos que haya a partir de la dirección1.

—ABRE "nombrearchivo (sólo admite disco)

o

ABRE "dispositivo.

Abre el archivo o el dispositivo especificado para aceptar o enviar caracteres.

Normalmente puede haber varios archivos o dispositivos abiertos al mismo tiempo. Si tecleamos:

?ABRE "TEXTO

pueden ocurrir dos cosas:

— Si el archivo TEXTO no existe, se creará uno con dicho nombre.

— Si el archivo TEXTO ya existe, se abrirá para lectura o escritura en ese momento.

Para abrir un dispositivo, por ejemplo, la impresora:

?ABRE "LPT1

Antes de apagar el ordenador recuerda que debes cerrar todos los archivos que hayas abierto.

NOTA: El LOGO del SPECTRUM no dispone de esta primitiva.

PONCURSOR no admite valores más grandes al número de filas ni de columnas que tenga la pantalla.

EXPERIENCIA Y PRACTICAS EN LOGO

—Cierra "nombrearchivo" (sólo admite disco)

o

CIERRA "dispositivo"

Cierra el archivo o el dispositivo que indiquemos.

Si tenemos abierto el archivo TEXTO, antes de desconectar el ordenador, o cuando no lo necesitemos, lo cerramos tecleando:

?CIERRA "TEXTO"

—CIERRATODO (sólo admite disco)

Cierra todos los archivos y dispositivos que se encuentren abiertos en ese momento.

—PONLECTURA "dispositivo"

El archivo o dispositivo que especifiquemos será fijado como fuente de datos para las primitivas LEECAR, LEECARS, LEEPALABRA Y LEELISTA.

El puntero de lectura se colocará al principio del archivo en cuanto tecleemos esta primitiva y el archivo a leer.

El dispositivo por defecto es el teclado, si tras fijar para lectura otro dispositivo queremos volver al inicial, tecleamos:

?PONLECTURA "CON"

o

?PONLECTURA []

Ten en cuenta que sólo podrás abrir para lectura aquellos dispositivos que lo permitan, como el teclado, y no podrás hacerlo, por ejemplo, con la impresora.

NOTA: El LOGO del SPECTRUM no dispone de esta primitiva.

—PONESCRITURA "nombrearchivo" (sólo admite disco)

o

PONESCRITURA "dispositivo"

El archivo o dispositivo que indiquemos será fijado como destino de los datos de ESCRIBE, MUESTRE Y TECLEA.

El puntero se colocará al final del archivo especificado.

Si hay más de un archivo o dispositivo abierto para escritura, los datos de las tres pri-

mitivas mencionadas anteriormente se escribirán en el último que haya sido abierto.

Por defecto, los datos se escriben en la pantalla. Si una orden PONESCRITURA desactiva este dispositivo, podemos volver a activarlo tecleando:

?PONESCRITURA "CON"

o

?PONESCRITURA []

Para abrir la impresora como dispositivo de escritura de datos, haríamos lo siguiente:

?ABRE "LPT1"

?PONESCRITURA "LPT1"

NOTA: El LOGO del SPECTRUM no dispone de esta primitiva.

—LECTURA

Devuelve la unidad y el nombre del archivo que esté abierto para lectura en ese momento.

?ESCRIBE LECTURA

B: TEXTO

El archivo TEXTO que se encuentra en la unidad B está abierto para lectura.

NOTA: La respuesta que hemos indicado ("B:TEXTO") es en el caso de los PC-COMPATIBLES, en los MSX la unidad en la que se encuentra en archivo para la lectura se representa por un número.

Esta primitiva no la contiene el LOGO del SPECTRUM.

—ESCRITURA

Devuelve el nombre del archivo o el dispositivo abierto para escritura en ese momento.

Si tecleamos:

?ESCRIBE ESCRITURA

la respuesta se escribirá en el archivo o en el dispositivo activo para escritura en ese momento.

NOTA: El LOGO del SPECTRUM no dispone de esta primitiva.

—FINLEC?

Abreviación de FIN de LECTura.

Podemos establecer el número de columnas que queramos que tenga la pantalla.

Devuelve CIERTO si el puntero está al final del archivo abierto en lectura en ese momento, si no, devuelve FALSO.

Si en el momento que tecleamos:

?ESCRIBE FINLEC?

no hay ningún archivo abierto para lectura, el LOGO mostrará el mensaje:

ARCHIVO NO ELEGIDO

NOTA: El LOGO del SPECTRUM no dispone de esta primitiva.

—FINESC?

Abreviación de FIN de ESCRITURA.

Devuelve CIERTO si el puntero está al final del archivo abierto en escritura en ese momento, si no, devuelve FALSO.

Como ocurre con FINLEC?, si no hay ningún archivo abierto para escritura en el momento que tecleamos:

?ESCRIBE FINESC?

el LOGO mostrará el mensaje:

ARCHIVO NO ELEGIDO

NOTA: El LOGO del SPECTRUM no dispone de esta primitiva.

■ Cuadro resumen

—?CURSOR

Devuelve dos números que corresponden a la columna y a la fila, respectivamente, de la posición del cursor.

—PONANCHURA [F F] (Para los MSX)

Sitúa una ventana de texto en la pantalla gráfica que va desde la fila que se indica con el primer número hasta la fila que se indica con el segundo.

—PONANCHURA n (Para los PC-Compatibles)

Establece el número de columnas en la pantalla que indica n.

—ANCHURA

MSX

Devuelve dos números que corresponden a la primera y última fila de la ventana de texto.

PC-COMPATIBLES

Devuelve un número indicando el número de columnas que tiene la pantalla.

—ARCHIVO? "nombrearchivo"

Devuelve CIERTO si existe el archivo indicado, si no, devuelve FALSO.

Esta primitiva sólo gestiona discos.

—CARGAB "nombrearchivo dirección"

Carga en memoria y a partir de la dirección especificada el archivo "nombrearchivo".

En la versión del SPECTRUM:

.CARGAB "nombrearchivo dirección"

—GARDAB "nombrearchivo dirección1 dirección2"

Guarda en un archivo los bytes comprendidos entre dirección1 y dirección2.

En el LOGO del SPECTRUM:

.GUARDAB "nombrearchivo dirección n"

Guarda en el archivo especificado los n octetos que halla a partir de dirección.

—ABRE "nombrearchivo (sólo disco)"

ABRE "dispositivo"

Abre el archivo o el dispositivo especificado para enviar o aceptar caracteres.

Si abrimos un fichero que no existe, éste se creará en ese momento.

El LOGO del SPECTRUM no dispone de esta primitiva.

—CIERRA "nombrearchivo (sólo disco)"

CIERRA "dispositivo"

Cierra el archivo o el dispositivo especificado.

Recuerda cerrar siempre todos los archivos abiertos antes de apagar el ordenador.

—CIERRRA TODO (sólo disco)

Cierra todos los archivos y dispositivos que estén abiertos en ese momento.

—PONLECTURA "nombrearchivo (sólo disco)"

PONLECTURA "dispositivo"

Fija como fuente de datos, para las primitivas LEECAR, LEECARs, LEEPALABRA Y LEELISTA, el archivo o dispositivo especificado.

PONCURSOR actúa sobre la primera orden escribe que se encuentra a continuación.

EXPERIENCIA Y PRACTICAS EN LOGO

El dispositivo abierto en lectura por defecto es el teclado.

El LOGO del SPECTRUM no dispone de esta primitiva.

—PONESCRIPTURA "nombrearchivo
(sólo disco)

PONESCRIPTURA "dispositivo

El archivo o dispositivo que especifiquemos será el destino de los datos de ESCRIBE, MUESTRA y TECLEA.

El dispositivo abierto en escritura, por defecto, es el teclado.

El LOGO del SPECTRUM no dispone de esta primitiva.

—LECTURA

Devuelve la unidad y el nombre del archivo que esté abierto en lectura en ese momento.

En los PC-COMPATIBLES la unidad es una letra (A, B...).

En los MSX la unidad es un número (1, 2...).

El LOGO del SPECTRUM no dispone de esta primitiva.

—ESCRITURA

Devuelve el nombre del archivo o el dispositivo abierto en escritura en este momento.

El LOGO del SPECTRUM no dispone de esta primitiva.

—FINLEC?

Devuelve CIERTO si el puntero está al final del archivo abierto en lectura en ese momento, si no, devuelve FALSO.

El LOGO del SPECTRUM no dispone de esta primitiva.

—FINESC:

Devuelve CIERTO si el puntero está al final del archivo abierto en escritura en ese momento, si no, devuelve FALSO.

El LOGO del SPECTRUM no dispone de esta primitiva, tampoco el LOGO del MSX.

■ Ejercicios

1. Dibuja la siguiente señal de tráfico.

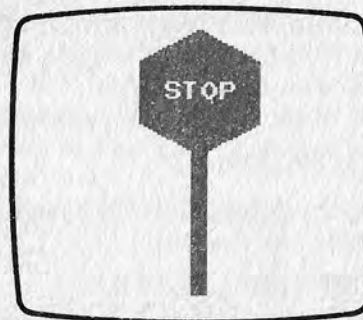


Fig. 1.

2. Dibuja cuatro marcadores y consigue que al pulsar 1 cuente el primero y pare lo que estaba haciendo, que al pulsar 2 lo haga el segundo, que al pulsar 3 lo haga el tercero y que al pulsar 4 lo haga el cuarto.

Determina también una tecla para finalizar la ejecución, por ejemplo, la F.

3. Intenta realizar el siguiente dibujo:

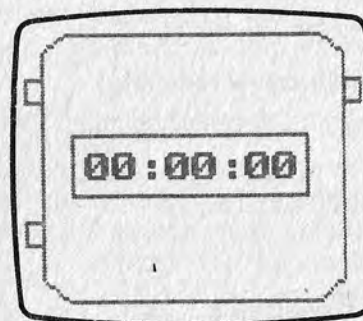


Fig. 2.

4. Define los procedimientos necesarios para, una vez dibujado el siguiente cronómetro, lo podamos poner en marcha y marque las décimas de segundo, los segundos y los minutos:

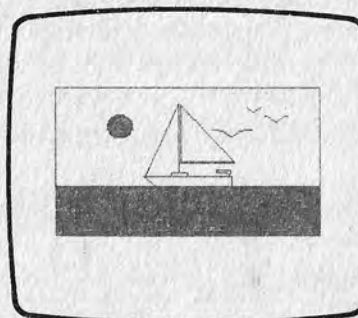


Fig. 3.

Si se dispone de un disco, el destino de los datos de las órdenes ESCRIBE, TECLEA Y MUESTRA puede ser un archivo.

5. ¿Son correctas las siguientes órdenes?:

```
-?PONCURSOR [90 1]
  ?ES "TORTUGA

-?PARA A
  >PONCURSOR [25 10]
  >ES "TROTAMUNDOS
  >FIN

-?PONANCHURA 10
  ?PONCURSOR [15 5]

-?ESCRIBE ARCHIVO? SEÑAL.LOG

-?ESCRIBE FINLEC?

-?PARA E3
  >PONCURSOR [20 20]
  >ESCRIBE CURSOR
  >FIN
```

```
>PONRUMBO 180
>BL
>REPITE 2 [AV 70 GI 90 AV 6 GI 90]
>SL PONPOS [7 0]
>BL
>RELLENA
>FIN
```

LETRA escribe la palabra STOP dentro de la señal.

```
?PARA LETRA
>PONCURSOR [19 8]
>ESCRIBE "STOP
>FIN
```

■ Solución a los ejercicios

1:

Definimos un procedimiento para inicializar la pantalla, otro para dibujar la señal y otro para escribir la palabra STOP.

SEÑAL inicializa la pantalla y llama a los dos procedimientos restantes.

```
>?PARA SEÑAL
>BP PONRUMBO 0
>FOR
>LETRA
>FIN
```

FOR dibuja la señal

```
?PARA FOR
>SL PONPOS [-18 25]
>BL
>PONCL 3
>REPITE 6 [AV 30 GD 60]
>SL PONPOS [-14 27]
>BL
>REPITE 6 [AV 26 GD 60]
>SL PONPOS [0 25]
>BL
>PONCL 2
>RELLENA
>SL PONPOS [-16 27]
>BL
>PONCL 3
>RELLENA
>SL PONPOS [5 11]
```

2:

El primer procedimiento que definimos (DIBUJO) inicializa la pantalla, llama cuatro veces al procedimiento que dibuja el marcador, y una vez al que realiza la cuenta.

```
?PARA DIBUJO
>BP
>OT
>SL
>PONPOS [-90 30]
>MARCADOR
>SL PONPOS [15 30]
>MARCADOR
>SL PONPOS [-90 -30]
>MARCADOR
>SL PONPOS [15 -30]
>MARCADOR
>PONCURSOR [9 6]
>ESCRIBE "MARCADOR$ 1
>PONCURSOR [22 6]
>ESCRIBE "MARCADOR$ 2
>PONCURSOR [9 12]
>ESCRIBE "MARCADOR$ 3
>PONCURSOR [22 12]
>ESCRIBE "MARCADOR$ 4
>HAZ "N LEECAR
>CUENTAS 0 0 0 0 :N
>FIN

MARCADOR dibuja un marcador:
?PARA MARCADOR
>BL
>PONCL 1
>REPITE 2 [AV 25 GD 90 AV 80 GD 90]
>SL
```

Utiliza la primitiva ARCHIVO ? para comprobar si existe o no un archivo especificado.

EXPERIENCIA Y PRACTICAS EN LOGO

```
>PONX COORX + 2
>PONY COORY + 2
>BL
>PONCL 2
>RELLENA
>FIN
```

CUENTAS es el encargado de parar y poner en marcha los marcadores correspondientes.

```
:A para el contador del marcador 1
:B para el contador del marcador 2
:C para el contador del marcador 3
:D para el contador del marcador 4
:N carácter pulsado
?PARA CUENTAS :A :B :C :D :N
>SI TECLA? [HAZ "N LEECAR]
>SI :N = 1 [PONCURSOR [10 8] HAZ "A
:A + 1 ESCRIBE :A]
>SI :N = 2 [PONCURSOR [23 8] HAZ "B
:B + 1 ESCRIBE :B]
>SI :N = 3 [PONCURSOR [10 14] HAZ "C
:C + 1 ESCRIBE :C]
>SI :N = 4 [PONCURSOR [23 14] HAZ "D
:D + 1 ESCRIBE :D]
>SI :N = "F [ALTO] [CUENTAS :A :B :C
:D :N]
```

El dibujo de los cuatro marcadores es el siguiente:



Fig. 4.

3:

Para hacer el dibujo, vamos a definir un procedimiento para cada una de las partes de que consta.

En primer lugar, definimos el que va a dibujar el marco y el mar:

```
?PARA MARCO
>SL
>PONPOS [-120 -60]
>BL
```

```
>REPITE 2 [AV 150 GD 90 AV 240 GD
90]
>SL
>PONPOS [-120 -10]
>GD 90
>BL
>AV 240
>SL
>GD 135 AV 2
>BL
>RELLENA
>FIN
```

Para dibujar el barco definimos varios procedimientos:

```
?PARA CASCO
>SL
>PONPOS [40 -10]
>BL
>PONRUMBO 0
>AV 10 GI 90
>AV 80 GI 135
>AV 15 GI 45
>AV 69
>SL
>PONPOS [0 0]
>PONRUMBO 0
>BL
>REPITE 2 [AV90 AV 15 GI 90]
>FIN
```

```
?PARA MASTIL
>SL
>PONPOS [-8 4]
>PONRUMBO 0
>BL
>AV 70 RE 60
>GD 90 AV 50
>SL
>PONPOS [-8 74]
>PONRUMBO -157
>BL
>AV 80
>FIN
```

```
?PARA VELA
>SL
>PONPOS [-6 15]
>BL
>PONRUMBO 0
>AV 58
>PONRUMBO 140
>AV 75
```

Podemos cargar en memoria archivos de tipo binario.

```
>PONRUMBO 270
>AV 48
>FIN
```

```
?PARA TIMON
>SL
>PONPOS [35 0]
>PONRUMBO 10
>BL
>AV 4
>PONRUMBO 270
>AV 10 GD 90
>AV 13 GD 100
>AV 6
>FIN
```

Ahora los agrupamos todos en uno:

```
?PARA BARCO
>CASCO
>MASTIL
>VELA
>TIMON
>FIN
```

El Sol lo dibuja el siguiente procedimiento:

```
?PARA SOL
>SL
>PONPOS [-50 50]
>BL
>REPITE 36 [AV 2 GD 10]
>SL
>GD 90 AV 3
>BL
>RELLENA
>FIN
```

Para dibujar los pájaros definimos tres procedimientos:

```
?PARA PAJAROS
>SL
>PONPOS [40 40]
>HAZ "T 10
>ALAD
>SL
>PONPOS [40 40]
>ALAI
>SL
>PONPOS [60 65]
>HAZ "T 4
>ALAD
>SL
>PONPOS [60 65]
```

```
>ALAI
>SL
>PONPOS [80 55]
>HAZ "T 6
>ALAD
>SL
>PONPOS [80 55]
>ALAI
>FIN
```

Este procedimiento realiza llamadas al procedimiento que dibuja el ala derecha (ALAD) y al que dibuja el ala izquierda (ALAI). La variable T es la que determina el tamaño de cada pájaro:

```
?PARA ALAD
>BL
>PONRUMBO 45
>REPITE :T [AV 2 GD 5]
>FIN

?PARA ALAI
>BL
>PONRUMBO -45
>REPITE :T [AV 2 GI 5]
>FIN
```

Una vez que tenemos todos los procedimientos necesarios definidos, definimos uno que inicializa el proceso y engloba a todos los demás:

```
?PARA DIBUJO
>BP
>OT
>MARCO
>BARCO
>SOL
>PAJAROS
>FIN
```

Para ver el dibujo, introducimos:
?DIBUJO

4:

En primer lugar, definimos el procedimiento que dibuja el cronómetro:

```
?PARA CRONOMETRO
>BP
>OT
>BT
>SL
>PONPOS [-40 -20]
```

Podemos guardar en binario un bloque de la memoria.

EXPERIENCIA Y PRACTICAS EN LOGO

```
>BL
>REPITE 4 [GI 45 AV 5 GD 45 AV 75 GD
  45 AV 5 GD 45]
>SL
>PONPOS [-35 11]
>BL
>REPITE 2 [AV 20 GD 90 AV 70 GD 90]
>SL
>PONPOS [-44 42]
>BOTON
>SL
>PONPOS [-44 -5]
>BOTON
>SL
>PONPOS [51 42]
>BOTON
>PONCURSOR [16 10]
>ES "00:00:00
>FIN
```

Este procedimiento realiza tres llamadas a BOTON, que es el que dibuja los tres botones del cronómetro:

```
?PARA BOTON
>PONRUMBO 270
>BL
>REPITE 2 [AV 5 GD 90 AV 8 GD 90]
>FIN
```

Una vez que tenemos el cronómetro dibujado, tenemos que definir tres procedimientos, uno para calcular las décimas de segundo, otro para calcular los segundos y otro para calcular los minutos:

DEC es un procedimiento recursivo que va incrementando el valor de la variable A desde 0 hasta 60. El valor de esta variable se va escribiendo en el lugar correspondiente a las décimas de segundo, cuando alcanza el valor 60, devuelve el control al procedimiento que lo ha llamado.

```
?PARA DEC
>SI TECLA? [ALTO]
>HAZ "A :A + 1
>SI :A = 60 [ALTO]
>SI :A > 9[PONCURSOR [22 10]]
  [PONCURSOR [23 10]]
>ES :A
>DEC
>FIN
```

SEG es el procedimiento que va calculando los segundos. Al igual que el anterior, una variable (B) es la que se va incrementando hasta alcanzar el valor 60. El valor de esta variable se escribe en el lugar correspondiente a los segundos.

Este procedimiento lo primero que realiza es una inicialización de la variable que se utiliza para las décimas de segundo y a continuación llama al procedimiento DEC.

DEC se hará sesenta veces y cuando termine devolverá el control a SEG. De esta forma, los segundos se incrementan en uno cada vez que se realiza sesenta veces DEC.

Este procedimiento también se realizará sesenta veces:

```
?PARA SEG
>SI TECLA? [ALTO]
>HAZ "A 0
>DEC
>HAZ "B :B + 1
>SI :B = 60 [PONCURSOR [19 10]
  ES "00 ALTO]
>SI :B > 9 [PONCURSOR [19 10]]
  [PONCURSOR [20 10]]
>ES :B
>SEG
>FIN
```

Por último, MIN realiza el mismo proceso que los anteriores, pero en este caso la variable que marca los minutos (C) se incrementará en uno cada vez que realice sesenta veces SEG:

Este procedimiento, en primer lugar, inicializa la variable de los segundos y a continuación llama al procedimiento SEG:

```
?PARA MIN
>HAZ "B 0
>SEG
>SI TECLA? [ALTO]
>HAZ "C :C + 1
>SI :C = 61 [ALTO]
>SI :C > 9 [PONCURSOR [16 10]]
  [PONCURSOR [17 10]]
>ES :C
>FIN
```

Los tres procedimientos hacen al principio la pregunta:

```
?SI TECLA? [ALTO]
```

Si no recuerdas qué archivo está abierto en lectura o en escritura en un momento determinado consúltaselo al Logo.

Esto lo hacemos para detener el cronómetro en el momento que queramos con solo pulsar una tecla.

También en los tres preguntamos si el valor de la variable correspondiente es mayor que 9. Esto lo hacemos para que cuando la variable tenga dos dígitos, el cursor se coloque en una columna anterior.

El procedimiento CRONO es el que inicializa el proceso y el que pone en marcha el cronómetro.

```
?PARA CRONO
>CRONOMETRO
>HAZ "A 0
>HAZ "B 0
>HAZ "C 0
>PONCURSOR [9 4]
>ES "PULSA$ M$ PARA$ COMENZAR
>HAZ "T LC
>SI :T = "M [PONCURSOR [7 17]
    ES "PULSA$ UNA$ TECLA$ PARA$
    PARAR MIN] [CRONO]
>FIN
```

Para ejecutarlo, bastará con introducir:

?CRONO

y en la pantalla aparecerá lo siguiente:



Fig. 5.

Una vez que pulsemos M, el cronómetro se pondrá en funcionamiento:



Fig. 6.

Podemos pararlo en cualquier momento pulsando cualquier tecla.

Para volver a ponerlo en funcionamiento tenemos que introducir de nuevo:

?CRONO

5:

```
—?PONCURSOR [90 1]
:ES "TORTUGA
```

INCORRECTA

El valor que se da para la columna es superior al número existente de ellas:

```
—?PARA A
>PONCURSOR [25 10]
>ESCRIBE "TROTAMUNDOS
>FIN
```

CORRECTA

La palabra que se comenzaría a escribir en la columna 25 y en la fila 10.

En esta fila aparecería:

TROTAMU

y al principio de la fila 11:

NDOS

```
—?PONANCHURA 10
```

```
?PONCURSOR [15 5]
```

INCORRECTA

Si se ha marcado una anchura de 10 columnas, el cursor no puede colocarse en una columna superior a 10.

```
—?ESCRIBE ARCHIVO? SEÑAL.LOG
INCORRECTO
```

Faltan las comillas delante del nombre del archivo.

El LOGO muestra el mensaje de error:

NO SE COMO HACER PARA SEÑAL.LOG

```
—?ESCRIBE FINLEC
```

INCORRECTO

Falta un interrogante detrás de FINLEC
El LOGO mostrará el mensaje:

NO SE COMO HACER PARA FINLEC

```
—?PARA E3
```

```
>PONCURSOR [20 20]
```

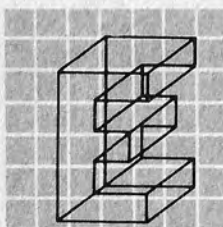
```
>ESCRIBE CURSOR
```

```
>FIN
```

CORRECTO

El cursor se posicionará en la fila 20, columna 20, y en ese lugar de la pantalla escribirá la posición actual del cursor (20 20).

MANEJO DE SPRITES Y ELEMENTOS GRAFICOS



L COMMODORE 64 es uno de los ordenadores con más capacidad gráfica y con más colores del mercado. Lo único que tiene de malo es que no dispone de ninguna sentencia o comando que

los controle; por ello, su estudio y utilización se hacen más difíciles y complejos.

Para empezar, comenzaremos viendo la tabla de los colores que podemos utilizar con este ordenador:

Color 0 — Negro
Color 1 — Blanco
Color 2 — Rojo
Color 3 — Turquesa

Color 4 — Malva/violeta
Color 5 — Verde
Color 6 — Azul
Color 7 — Amarillo
Color 8 — Naranja
Color 9 — Marrón
Color 10 — Rojo claro
Color 11 — Gris 1
Color 12 — Gris 2
Color 13 — Verde claro
Color 14 — Azul claro
Color 15 — Gris 3

Estos son todos los colores que podemos utilizar en el COMMODORE. Para ver cómo aparecen por la pantalla, vamos a ver el programa 1:

```

100 REM *****
110 REM * PROGRAMA DEMO PARA VER *
120 REM * TODOS LOS COLORES DEL *
130 REM * COMMODORE 64. *
140 REM *****
150 REM
160 REM
170 REM *** LECTURA DE LAS LINEAS DE DATA ***
180 REM
190 DIM A$(15)
200 RESTORE
210 FOR I=0 TO 15
220   READ A$(I)
230 NEXT I
240 REM
250 REM *****
260 REM * DATAS CON LOS NOMBRES DE LOS COLORES *
270 REM *****
280 REM
290 DATA "Negro","Blanco","Rojo oscuro","Turquesa"
300 DATA "Malva o Violeta","Verde oscuro","Azul oscuro"
310 DATA "Amarillo","Naranja","Marron","Rojo claro","Gris 1"
320 DATA "Gris 2","Verde claro","Azul claro","Gris 3"
    
```

```

330 REM
340 REM *****
350 REM * PROGRAMA PRINCIPAL *
360 REM *****
370 REM
380 LET C=PEEK(53280!)
390 FOR I=0 TO 1 STEP 0
400   FOR J=0 TO 15
410     PRINT CHR$(147)
420     PRINT:PRINT
430     PRINT "EL COLOR DEL BORDE ES ";A$(J)
440     FOR K=1 TO 10
450       PRINT
460     NEXT K
470     PRINT "PULSA 'P' PARA PARAR"
480     POKE 53280!,J
490     FOR K=1 TO 400
500       GET B$
510       IF B$<>" " THEN GOTO 530
520     NEXT K
530     IF B$="P" OR B$="p" THEN GOTO 560
540   NEXT J
550 NEXT I
560 PRINT CHR$(147)
570 POKE 53280!,C
580 END

```

Como podemos ver en el programa, el color del borde de la pantalla va cambiando de color hasta que pulsamos la letra "p". Antes de continuar, tenemos que ver algo más sobre la pantalla.

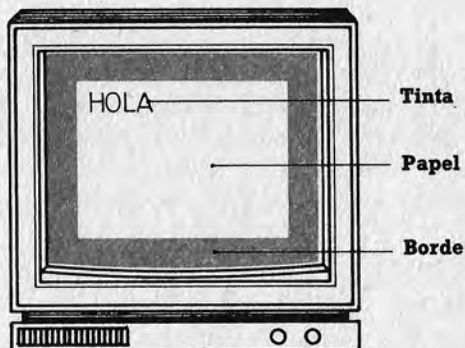


Fig. 1.

Si consideramos la pantalla del ordenador como una hoja de papel, estaremos en unas condiciones muy buenas para entender cómo funciona y cómo podemos trabajar con el color en la pantalla. Lo primero que tenemos que saber es que dicha hoja de papel no ocupa toda la pantalla, sino sólo la parte central. El resto, la parte que está rodeando a la hoja de papel, es lo que se llama BORDE. No podemos escribir ni dibujar nada en el borde

pero sí podemos hacerle cambiar de color. Esto es lo que hacemos en el programa 1.

Para cambiar el color del borde sólo tenemos que cambiar el contenido de una posición de memoria mediante el uso de la sentencia POKE.

Por si el lector ha olvidado o no conoce esta instrucción, vamos a ver en unas pocas líneas para qué sirve.

La sentencia POKE nos sirve para acceder directamente a la memoria del ordenador y escribir datos en ella. Para ello le tenemos que decir en qué dirección queremos escribir y qué valor vamos a almacenar. Su sintaxis es:

POKE dirección de memoria, valor

La dirección de memoria es un número comprendido entre 0 y 65535. Este último número no es arbitrario, pues nuestro ordenador tiene 64 Kbytes de memoria. Como todos sabemos que un Kbyte es igual que 1024 bytes, si multiplicamos 64×1.024 nos da 65.536. Como las posiciones de memoria se numeran empezando por la cero (0), eso nos dice que la posición más alta de memoria que tiene el ordenador es la 65.535.

El valor que queremos introducir en dicha dirección de memoria ha de ser un número entre 0 y 255. Esto tampoco es arbitrario. La

MANEJO DE SPRITES Y ELEMENTOS GRAFICOS

razón a la que obedece es que como nuestro ordenador trabaja con bytes y cada byte tiene ocho bits, eso nos dice que cada byte puede tener un valor entre 0 y 255 como máximo.

Así pues, una vez repasado el uso de la sentencia POKE, podemos seguir con el uso del color. Decíamos que el programa 1 cambiaba el color del borde cambiando el contenido de una posición de memoria. Esta posición de memoria es la 53.280 y en ella tendremos que poner el código del color que queremos que tenga el borde. Por ejemplo, si queremos que el borde sea de color rojo oscuro, tendremos que poner:

POKE 53280,2

y si lo queremos de color azul claro, tendremos que poner:

POKE 53280,14

Una vez entendido esto podemos entender mejor el programa 1. En él realizamos un bucle desde 0 hasta 15, y en la línea 480 pokeamos el valor de la variable índice de dicho bucle en la dirección indicada. Con ello conseguimos que el color del borde vaya cambiando continuamente.

Volvamos ahora a fijarnos en la pantalla del ordenador. Hemos dicho que podríamos considerar que la parte en la cual podemos imprimir es un papel. Lógicamente, como ocurre con todos los papeles, podremos escribir en él lo que queramos. Según esto, y sabiendo que nuestro COMMODORE tiene 16 colores distintos, es lógico pensar que el color del papel puede ser de cualquiera de esos 16 colores. Lógicamente, también podemos cambiar el color de la tinta de los caracteres.

Cambiar al color del papel no es sencillo, por ello lo vamos a dejar para un poco más adelante. Para cambiar el color de la tinta es mucho más fácil y se puede realizar de tres formas distintas. Estas son:

1. Incluir los códigos de color dentro de las comillas de las sentencias PRINT.
2. Especificar qué color queremos mediante el uso de la función CHR\$().
3. POKEar en la memoria de color.

Cada una de estas tres formas es buena para ocasiones distintas. Veamos cómo realizar cada una de ellas.

La forma más rápida y más sencilla de cambiar el color de las letras es incluir los órdenes de color dentro de las comillas de los comandos PRINT. Para ello, al igual que hace-

mos con el movimiento del cursor y borrado de pantalla, tenemos que pulsar una serie de teclas. En la figura 1 podemos ver qué combinación de teclas nos da cada color y cómo aparece dicho color representado entre las comillas del PRINT.

TECLA	COLOR	SE VE COMO
CTRL-1	NEGRO	
CTRL-2	BLANCO	
CTRL-3	ROJO	
CTRL-4	CYAN	
CTRL-5	PURPURA	
CTRL-6	VERDE	
CTRL-7	AZUL	
CTRL-8	AMARILLO	
CMB-1	NARANJA	
CBM-2	MARRON	
CBM-3	ROJO CLARO	
CBM-4	GRIS 1	
CBM-5	GRIS 2	
CBM-6	VERDE CLARO	
CBM-7	AZUL CLARO	
CBM-8	GRIS 3	

Fig. 2.

Hace falta explicar un poco el significado de la tabla. Donde pone CTRL significa CONTROL. Por ejemplo, para obtener el color blanco tenemos que pulsar la tecla CONTROL y, sin soltarla, pulsar la tecla 2. Con ello nos aparecerá en la pantalla una "E" mayúscula invertida. Cualquier mensaje que imprimamos a continuación aparecerá en blanco en cuanto ejecutemos la sentencia PRINT.

PRINT "<CTRL-2> ESCRITO EN BLANCO"

A partir del color naranja, para obtener cada color hay que pulsar una combinación distinta de teclas. En vez de pulsar la tecla CONTROL tenemos que pulsar la tecla CBM. Esta tecla es la que lleva el anagrama de COMMODORE dibujado. El resto es igual que si hubiésemos pulsado la tecla CONTROL.

El programa que vamos a ver a continuación nos muestra cómo aparecen todos los colores de tinta en la pantalla. Cada vez que en listado aparezca algo entre corchetes, significa que lo que hay dentro no hay que escribirlo, sino pulsar las teclas que indica.

```

100 REM *****
110 REM * PROGRAMA DEMO PARA VER COMO *
120 REM * HAY QUE CAMBIAR LA TINTA EN *
130 REM * EL COMMODORE 64.          *
140 REM *****
150 REM
160 REM *** LECTURA DE LAS LINEAS DE DATA ***
170 REM
180 DIM A$(15)
190 RESTORE
200 FOR I=0 TO 15
210   READ A$(I)
220 NEXT I
230 REM
240 REM *****
250 REM * DATAS CON LOS NOMBRES DE LOS COLORES *
260 REM *****
270 REM
280 DATA "Negro","Blanco","Rojo oscuro","Turquesa"
290 DATA "Malva o Violeta","Verde oscuro","Azul oscuro"
300 DATA "Amarillo","Naranja","Marron","Rojo claro","Gris 1"
310 DATA "Gris 2","Verde claro","Azul claro","Gris 3"
320 REM
330 REM *****
340 REM * PROGRAMA PRINCIPAL *
350 REM *****
360 REM
370 LET B$="ESTE MENSAJE ES DE COLOR "
380 GOSUB 850
390 PRINT "<CTRL-1>";B$;A$(0)
400 GOSUB 850
410 PRINT "<CTRL-2>";B$;A$(1)
420 GOSUB 850
430 PRINT "<CTRL-3>";B$;A$(2)
440 GOSUB 850
450 PRINT "<CTRL-4>";B$;A$(3)
460 GOSUB 850
470 PRINT "<CTRL-5>";B$;A$(4)
480 GOSUB 850
490 PRINT "<CTRL-6>";B$;A$(5)
500 GOSUB 850
510 PRINT "<CTRL-7>";B$;A$(6)
520 GOSUB 850
530 PRINT "<CTRL-8>";B$;A$(7)
540 GOSUB 850
550 PRINT "<CBM-1>";B$;A$(8)
560 GOSUB 850
570 PRINT "<CBM-2>";B$;A$(9)
580 GOSUB 850
590 PRINT "<CBM-3>";B$;A$(10)
600 GOSUB 850
610 PRINT "<CBM-4>";B$;A$(11)
620 GOSUB 850
630 PRINT "<CBM-5>";B$;A$(12)
640 GOSUB 850
650 PRINT "<CBM-6>";B$;A$(13)
660 GOSUB 850
670 PRINT "<CBM-7>";B$;A$(14)

```

MANEJO DE SPRITES Y ELEMENTOS GRAFICOS

```

680 GOSUB 850
690 GOTO 380
700 PRINT "<CBM-7>";B$;A$(15)
710 REM
720 REM *****
730 REM * BORRADO DE PANTALLA, IMPRESION DE MENSAJE *
740 REM * Y COLOCACION DEL CURSOR EN HOME.          *
750 REM *****
760 REM
770 PRINT CHR$(147)
780 FOR I=1 TO 20
790   PRINT
800 NEXT I
810 PRINT "PULSA UNA TECLA"
820 PRINT CHR$(19)
830 PRINT:PRINT:PRINT
840 RETURN
850 REM
860 REM *****
870 REM * PULSA UNA TECLA *
880 REM *****
890 REM
900 GET C$
910 IF C$="" THEN GOTO 910
920 GOSUB 710
930 RETURN

```

La segunda manera que tenemos de cambiar el color de los caracteres es utilizando la función CHR\$. Para el COMMODORE, los colores no son sino caracteres; por ello, si decimos que nos imprima un carácter de los que definen un color, lo que escribamos a continuación será de dicho color. A continuación aparecen los colores seguidos de su número de carácter.

Color 0 — Negro —	CHR\$(144)
Color 1 — Blanco —	CHR\$(5)
Color 2 — Rojo —	CHR\$(28)
Color 3 — Cyan —	CHR\$(159)
Color 4 — Violeta —	CHR\$(156)
Color 5 — Verde —	CHR\$(30)
Color 6 — Azul —	CHR\$(31)

Color 7 — Amarillo —	CHR\$(158)
Color 8 — Naranja —	CHR\$(129)
Color 9 — Marrón —	CHR\$(149)
Color 10 — Rojo C. —	CHR\$(150)
Color 11 — Gris 1 —	CHR\$(151)
Color 12 — Gris 2 —	CHR\$(152)
Color 13 — Verde C. —	CHR\$(153)
Color 14 — Azul C. —	CHR\$(154)
Color 15 — Gris 3 —	CHR\$(155)

El programa que aparece a continuación nos muestra cómo podemos cambiar los colores utilizando la función CHR\$(). En definitiva, si lo que queremos es escribir la palabra HOLA en verde claro, lo único que tenemos que hacer es:

```
PRINT CHR$(153); "HOLA"
```

```

100 REM *****
110 REM * PROGRAMA DEMO PARA VER COMO *
120 REM * HAY QUE CAMBIAR LA TINTA EN *
130 REM * EL COMMODORE 64 UTILIZANDO *
140 REM * LA FUNCION CHR$().          *
150 REM *****
160 REM
170 REM *** LECTURA DE LAS LINEAS DE DATA ***
180 REM

```

```

190 DIM A$(15)
200 DIM A(15)
210 RESTORE
220 FOR I=0 TO 15
230   READ A$(I)
240   READ A(I)
250 NEXT I
260 REM
270 REM *****
280 REM * DATAS CON LOS NOMBRES DE LOS COLORES *
290 REM *****
300 REM
310 DATA "Negro",144,"Blanco",5,"Rojo oscuro",28,"Turquesa",159
320 DATA "Malva o Violeta",156,"Verde oscuro",30,"Azul oscuro",31
330 DATA "Amarillo",158,"Naranja",129,"Marron",149,"Rojo claro",150,"Gris 1",151
340 DATA "Gris 2",152,"Verde claro",153,"Azul claro",154,"Gris 3",155
350 REM
360 REM *****
370 REM * PROGRAMA PRINCIPAL *
380 REM *****
390 REM
400 FOR I=0 TO 1 STEP 0
410   FOR J=0 TO 15
420     PRINT CHR$(147)
430     PRINT:PRINT:PRINT
440     PRINT CHR$(A(J));"ESTE COLOR ES ";A$(J)
450     PRINT:PRINT:PRINT
460     PRINT "PULSA UNA TECLA"
470     GET B$
480     IF B$="" THEN GOTO 470
490   NEXT J
500   PRINT:PRINT:PRINT
510   PRINT "PULSA 'T' PARA TERMINAR"
520   GET B$
530   IF B$="" THEN GOTO 520
540   IF B$="T" OR B$="t" THEN GOTO 560
550 NEXT I
560 PRINT CHR$(147);CHR$(A(0))
570 END

```

Por fin, y ya para terminar con la tinta, vamos a ver la última forma de cambiar el color de la tinta. Para ello, primero tenemos que entender cómo está organizada la memoria de pantalla. A partir de la posición 1024 y hasta la 2023, ambas inclusive, el ordenador almacena el contenido de la pantalla que estamos visualizando. Por ejemplo, si escribimos:

POKE 1024,1

en la esquina superior derecha de la pantalla nos aparecerá una "A" mayúscula. La memoria se organiza tal y como aparece en la figura 3.

Como puede verse en el dibujo la primera línea de texto va desde la posición 1024

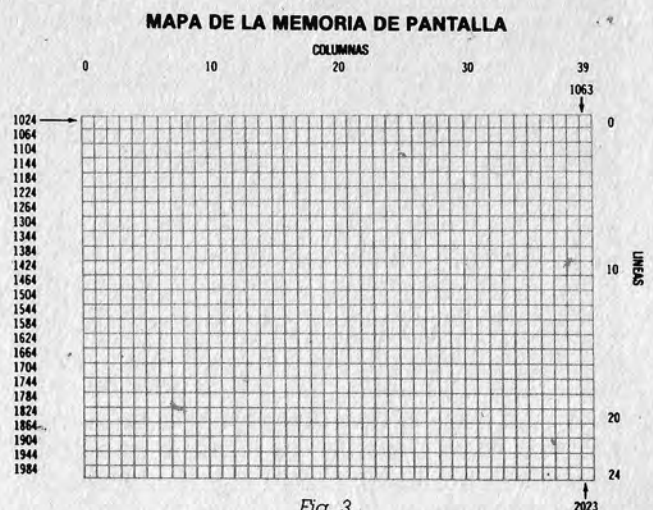


Fig. 3.

MANEJO DE SPRITES Y ELEMENTOS GRAFICOS

hasta la 1063. En la 1064 empieza la siguiente línea. El programa que aparece a continuación nos va a llenar toda la pantalla de "A" mayúsculas:

```
100 PRINT CHR$(147)
110 FOR I = 1024 TO 2023
120 POKE I,1
130 FOR J = 1 TO 100
140 NEXT J
150 NEXT I
160 END
```

Paralela a esta memoria de pantalla tenemos otra zona que es la memoria del color de la pantalla. Dicha zona empieza en la posición 55296 y también ocupa 1000 bytes, por lo que llega hasta la 56295. Si pokeamos valores entre 0 y 15 en dichas direcciones, el ordenador cambiará los colores de las letras que estén colocadas en la misma posición relativa a la memoria de pantalla. Así, si pokeamos el número 1 en la posición 55296

POKE 55296,1

la letra "A" que está colocada en la esquina superior izquierda de la pantalla, y que antes era

de color violeta, pasará a ser de color blanco. En la figura 4 aparece con más detalle la organización de la memoria de color.



Fig. 4.

El siguiente programa nos llena toda la pantalla de letras y luego nos cambia el color de las mismas con sólo pokear en la memoria.

```
100 REM *****
110 REM * PROGRAMA DEMO PARA VER COMO *
120 REM * HAY QUE CAMBIAR LA TINTA EN *
130 REM * EL COMMODORE 64 UTILIZANDO *
140 REM * LA MEMORIA DE COLOR. *
150 REM *****
160 REM
170 PRINT CHR$(147)
180 FOR I=1024 TO 2023
190 POKE I,1
200 NEXT I
210 LET K=0
220 FOR I=55296! TO 56295!
230 POKE I,K
240 LET K=K+1
250 IF K=16 THEN LET K=0
260 NEXT I
270 END
```

Habíamos comentado un poco más arriba que cambiar el color del papel en el COMMODORE era algo más complicado. De momento hay que decir que nunca pueden estar los 16 colores de que dispone este ordenador a la vez en la pantalla. Lo máximo que podemos conseguir es que haya cuatro colores distintos de fondo.

Para empezar, si lo que queremos es cambiar de golpe todo el color del papel de toda la pantalla, sólo tenemos que pokear en la dirección de memoria 53281 el color que queramos ver. Por ejemplo, si escribimos:

POKE 53281,0

la pantalla entera se pondrá de color negro.

Para poder usar los otros tres colores de fondo, tenemos que poner la pantalla del COMMODORE en el modo de fondo extendido. Para hacerlo tenemos que poner a uno el BIT número 6 de la posición de memoria 53265. Para hacerlo nos basta con teclear:

POKE 53265, PEEK(53265) OR 64

Para salir de este modo de pantalla y volver al modo normal (modo en el que se encuentra el COMMODORE cuando lo encendemos), tendremos que escribir:

POKE 53265, PEEK(53265) AND 191

Una vez que nos encontremos en el modo extendido de color de fondo, para poder decir qué cuatro colores queremos utilizar como fondo, tenemos que pokear los números de color en las posiciones 53281, 53282, 53283, 53284. En cualquiera de estas posiciones de memoria podemos poner un número entre 0 y 15.

Por otro lado, hay que decir que en este modo de pantalla no podemos utilizar más de 64 caracteres a la vez. Esto es debido a que el ordenador divide la tabla de caracteres en cuatro partes iguales y con los mismos carac-

teres. La única diferencia que existe entre las cuatro tablas es que, dependiendo de que el carácter que imprimamos sea de una tabla o de otra, el color de fondo será uno u otro. Por ello, si escribimos:

POKE 1024,1

nos aparecerá en la esquina superior izquierda de la pantalla una "A" mayúscula con el mismo color de fondo de la pantalla, pero si escribimos:

POKE 1024,65

nos vuelve a aparecer la letra "Z" mayúscula, pero esta vez el color de fondo es distinto. Si escribiésemos:

POKE 1024,1+64+64

la letra vuelve a aparecer, pero el color de fondo ha vuelto a cambiar. Esto es debido a que la letra "A" está repetida en las cuatro tablas en las que se ha dividido la tabla de caracteres inicial, pero con colores de fondo distintos.

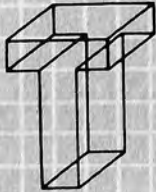
Ya para terminar vamos a ver una tabla que nos puede servir para saber qué combinaciones de colores son mejores en el COMMODORE.

COLOR DEL CARACTER

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	X	•	X	•	•	□	X	•	•	X	•	•	•	•	•	•
1	•	X	•	X	•	•	•	X	□	•	□	•	•	X	•	•
2	X	•	X	X	□	X	X	•	•	X	•	X	X	X	X	□
3	•	X	X	X	X	□	•	X	X	X	X	□	X	X	□	X
4	•	□	X	X	X	X	X	X	X	X	X	X	X	X	X	□
5	•	□	X	□	X	X	X	X	X	X	X	□	X	•	X	□
6	□	•	X	•	X	X	X	X	X	X	X	X	X	□	•	•
7	•	X	•	X	X	X	□	X	□	•	□	•	•	X	X	X
8	□	•	•	X	X	X	X	•	X	•	X	X	X	X	X	□
9	X	•	X	X	X	X	X	•	•	X	•	X	X	X	X	•
10	□	□	•	X	X	X	X	□	X	•	X	X	X	X	X	□
11	•	•	X	□	X	X	X	•	X	X	X	X	•	•	□	•
12	•	•	□	X	X	X	□	X	X	□	X	•	X	X	X	•
13	•	X	X	X	X	•	□	X	X	X	X	•	X	X	X	X
14	•	•	X	•	X	X	•	X	X	X	X	□	X	X	X	□
15	•	•	•	X	□	□	•	X	X	□	□	•	•	X	□	X

□ Excelente
• Buena
X Mala

TRUCOS Y RUTINAS BASICAS



ERMINANDO con la serie de programas que nos van a permitir definir nuestros propios caracteres gráficos y sprites, en este tomo vamos a dar un programa que nos permitirá definir caracteres en el COMMODORE

64. Con este programa podremos definir dos series distintas de caracteres: los que tenemos normalmente y un nuevo banco de caracteres por si el primero no fuese suficiente.

Para definir los caracteres utilizaremos una parrilla de 8×8 puntos dentro de la cual iremos iluminando los puntos que queremos

sean visibles en el carácter. Dentro de la parrilla veremos un cursor parpadeante. Para moverlo tendremos que pulsar las siguientes teclas:

- P — Movimiento hacia la derecha
- O — Movimiento hacia la izquierda
- Q — Movimiento hacia arriba
- A — Movimiento hacia abajo.

Una vez que hayamos colocado el cursor en el punto que queremos iluminar (o apagar) sólo tenemos que pulsar la barra espaciadora para hacerlo. Si dicho punto estuviese apagado, entonces se iluminaría. Si estuviese iluminado, se apagaría.

```
1000 REM *****
1010 REM * DEFINIDOR DE CARACTERES PARA EL COMMODORE 64 *
1020 REM *****
1030 REM
1040 POKE 650,128
1050 POKE 53281!,254
1060 PRINT CHR$(147)
1070 POKE 56334!,0
1080 POKE 1,51
1090 FOR K=0 TO 4095
1100   POKE 57344!+K,PEEK(53248!+K)
1110 NEXT K
1120 POKE 1,55
1130 POKE 56334!,1
1140 FOR I=0 TO 255
1150   POKE 50176!+I,I
1160 NEXT I
1170 DIM C(8),CA$(511,7)
1180 LET PA=1024
1190 POKE 53280!,1
1200 POKE 53281!,1
1210 PRINT CHR$(144);CHR$(147)
1220 LET CA=0
1230 LET PO=57344!
1240 LET JU=0
1250 GOSUB 1890
1260 GOSUB 2080
1270 LET X=0
1280 LET Y=0
1290 GOSUB 1490
1300 GOSUB 2370
1310 GET A$
1320 IF A$="" THEN GOTO 1310
1330 LET T=ASC(A$)
1340 IF T<>32 THEN GOTO 1370
1350 IF 2^(7-X) AND C(Y) THEN LET C(Y)=C(Y)-2^(7-X):GOTO 1370
1360 LET C(Y)=C(Y) OR 2^(7-X)
1370 IF T>132 AND T<141 THEN GOTO 1450
1380 LET T=PA+41+X+40*Y
1390 IF C(Y) AND 2^(7-X) THEN POKE T,81:GOTO 1410
```

```

1400 POKE T,46
1410 LET X=(X-(A$="P"))+(A$="O")) AND 7
1420 LET Y=(Y-(A$="A"))+(A$="Q")) AND 7
1430 GOSUB 1640
1440 GOTO 1310
1450 ON T-132 GOSUB 2600,2700,2900,3780,2800,3020,3170,1700
1460 GOTO 1310
1470 REM
1480 REM *****
1490 REM * ESCRIBE PARRILLA *
1500 REM *****
1510 REM
1520 PRINT CHR$(19);CHR$(145);
1530 FOR I=0 TO 7
1540     LET T=C(I)
1550     LET T1=128
1560     PRINT CHR$(29);
1570     FOR G=1 TO 8
1580         IF T AND T1 THEN PRINT CHR$(113);:GOTO 1600
1590         PRINT ". ";
1600         LET T1=T1/2
1610     NEXT G
1620     PRINT
1630 NEXT I
1640 LET T=PA+41+X+40*Y
1650 IF C(Y) AND 2^(7-X) THEN POKE T,209:RETURN
1660 POKE T,174
1670 RETURN
1680 REM
1690 REM *****
1700 REM * BORRA PARRILLA *
1710 REM *****
1720 REM
1730 FOR I=0 TO 7
1740     LET C(I)=0
1750 NEXT I
1760 GOSUB 1490
1770 RETURN
1780 REM
1790 REM *****
1800 REM * CAMBIO DE CHR$ A CODIGO DE PANTALLA *
1810 REM *****
1820 REM
1830 LET T=ASC(A$)
1840 IF T$>"?" AND T<96 THEN LET T=T-64:GOTO 1860
1850 IF T$>CHR$(95) THEN LET T=T-128
1860 RETURN
1870 REM
1880 REM *****
1890 REM * PINTA EL MARCO DE LA PARRILLA *
1900 REM *****
1910 REM
1920 PRINT CHR$(19);CHR$(111)
1930 FOR I=1 TO 8
1940     PRINT CHR$(163);
1950 NEXT I
1960 PRINT CHR$(112)
1970 FOR I=0 TO 7
1980     PRINT CHR$(116);"          ";CHR$(167)
1990 NEXT I
2000 PRINT CHR$(108);
2010 FOR I=1 TO 8
2020     PRINT CHR$(114);
2030 NEXT I
2040 PRINT CHR$(186)

```

TRUCOS Y RUTINAS BASICAS

```

2050 RETURN
2060 REM
2070 REM *****
2080 REM * PONE MENU PRINCIPAL *
2090 REM *****
2100 REM
2110 GOSUB 2460
2120 PRINT CHR$(19);
2130 FOR I=1 TO 13
2140     PRINT CHR$(145);
2150 NEXT I
2160 PRINT "      -----MENU PRINCIPAL-----";
2170 PRINT CHR$(145)
2180 PRINT CHR$(18);CHR$(31);"F1";CHR$(146);CHR$(144);
2190 PRINT "-INVIERTE CARACTER. "
2200 PRINT CHR$(18);CHR$(31);"F5";CHR$(146);CHR$(144);
2210 PRINT "-ESPEJO VERTICAL"
2220 PRINT CHR$(18);CHR$(31);"F2";CHR$(146);CHR$(144);
2230 PRINT "-ROTACION DERECHA. "
2240 PRINT CHR$(18);CHR$(31);"F6";CHR$(146);CHR$(144);
2250 PRINT "-SAVE/LOAD. "
2260 PRINT CHR$(18);CHR$(31);"F3";CHR$(146);CHR$(144);
2270 PRINT "-ROTACION IZQUIERDA. "
2280 PRINT CHR$(18);CHR$(31);"F7";CHR$(146);CHR$(144);
2290 PRINT "-JUEGOS/ASIGNA. "
2300 PRINT CHR$(18);CHR$(31);"F4";CHR$(146);CHR$(144);
2310 PRINT "-ESPEJO HORIZONTAL. "
2320 PRINT CHR$(18);CHR$(31);"F8";CHR$(146);CHR$(144);
2330 PRINT "-BORRAR. "
2340 RETURN
2350 REM
2360 REM *****
2370 REM * DATOS DE LA DEFINICION *
2380 REM *****
2390 REM
2400 PRINT CHR$(19);CHR$(145);CHR$(145);CHR$(145);TAB(11);"JUEGO:";JU
2410 PRINT TAB(11);"ULTIMO CARACTER:";CA;CHR$(157);"--";TAB(28);"-";CHR$(31);
;CHR$(144)
2420 POKE 1219,CA
2430 RETURN
2440 REM
2450 REM *****
2460 REM * BORRA LAS LINEAS DE LA PANTALLA *
2470 REM *****
2480 REM
2490 FOR X=12 TO 24
2500     POKE 781,X
2510     SYS 59903
2520 NEXT X
2530 PRINT CHR$(19);
2540 FOR G=0 TO 12
2550     PRINT CHR$(17);
2560 NEXT G
2570 RETURN
2580 REM
2590 REM *****
2600 REM * INVERSION DE UN CARACTER *
2610 REM *****
2620 REM
2630 FOR I=0 TO 7
2640     LET C(I)=ABS(C(I)-255)
2650 NEXT I
2660 GOSUB 1490
2670 RETURN
2680 REM

```

```

2680 REM *****
2700 REM * ROTACION DERECHA *
2710 REM *****
2720 REM
2730 FOR I=0 TO 7
2740   LET C(I)=(C(I)*2 AND 255)+(C(I) AND 128)/128
2750 NEXT I
2760 GOSUB 1490
2770 RETURN
2780 REM
2790 REM *****
2800 REM * ROTACION DERECHA *
2810 REM *****
2820 REM
2830 FOR I=0 TO 7
2840   LET C(I)=C(I)/2+(C(I) AND 1)*128
2850 NEXT I
2860 GOSUB 1490
2870 RETURN
2880 REM
2890 REM *****
2900 REM * ESPEJO VERTICAL *
2910 REM *****
2920 REM
2930 FOR I=0 TO 3
2940   LET T=C(I)
2950   LET C(I)=C(7-I)
2960   LET C(7-I)=T
2970 NEXT I
2980 GOSUB 1490
2990 RETURN
3000 REM
3010 REM *****
3020 REM * ESPEJO HORIZONTAL *
3030 REM *****
3040 REM
3050 FOR I=0 TO 7
3060   LET T=C(I)
3070   LET C(I)=0
3080   FOR G=0 TO 3
3090     IF T AND 2^G THEN LET C(I)=C(I)+2^(7-G)
3100     IF T AND 2^(7-G) THEN LET C(I)=C(I)+2^G
3110   NEXT G
3120 NEXT I
3130 GOSUB 1490
3140 RETURN
3150 REM
3160 REM *****
3170 REM * SAVE/LOAD *
3180 REM *****
3190 REM
3200 GOSUB 2460
3210 PRINT TAB(6); "----- MENU DE CASSETTE-----"
3220 PRINT
3230 PRINT CHR$(18);CHR$(31);"F1";CHR$(146);CHR$(144);
3240 PRINT "-SAVE JUEGO.";TAB(20);
3250 PRINT CHR$(18);CHR$(31);"F3";CHR$(146);CHR$(144);
3260 PRINT "-LOAD JUEGO."
3270 PRINT CHR$(18);CHR$(31);"F2";CHR$(146);CHR$(144);
3280 PRINT "-RETORNO AL MENU PRINCIPAL."
3290 GET A$
3300 IF A$="" THEN GOTO 3290
3310 LET A=ASC(A$)
3320 IF A=132 THEN LET O$="OPCION SAVE":GOTO 3360

```

TRUCOS Y RUTINAS BASICAS

```

3330 IF A=133 THEN LET O$="OPCION LOAD":GOTO 3360
3340 IF A=134 THEN GOSUB 2110:RETURN
3350 GOTO 3290
3360 PRINT
3370 PRINT TAB(8);"-----";O$;"-----"
3380 PRINT
3390 INPUT "NOMBRE ";T$
3400 IF T$="" OR LEN(T$)>10 THEN POKE 781,20:SYS 59903:PRINT CHR$(147):GOTO 3390
3410 REM
3420 REM *****
3430 REM * LOAD *
3440 REM *****
3450 REM
3460 IF A=132 THEN GOTO 3620
3470 PRINT "PULSE ALGUNA TECLA PARA EMPEZAR A CARGAR"
3480 POKE 1,PEEK(1) AND 39
3490 POKE 192,0
3500 GET A$
3510 IF A$="" THEN GOTO 3500
3520 OPEN 1,1,0,T$
3530 FOR I=JU*256 TO JU*256+255
3540   FOR G=0 TO 7
3550     INPUT #1,CAX(I,G)
3560   NEXT G
3570 NEXT I
3580 CLOSE 1
3590 GOTO 3200
3600 REM
3610 REM *****
3620 REM * SAVE *
3630 REM *****
3640 REM
3650 PRINT "PULSE ALGUNA TECLA PARA EMPEZAR A GRABAR"
3660 POKE 1,PEEK(1) AND 39
3670 POKE 192,0
3680 GET A$
3690 IF A$="" THEN GOTO 3680
3700 FOR I=JU*256 TO JU*256+255
3710   FOR G=0 TO 7
3720     PRINT #1,CAX(I,G)
3730   NEXT G
3740 NEXT I
3750 GOTO 3200
3760 REM
3770 REM *****
3780 REM * JUEGOS/ASIGNA *
3790 REM *****
3800 REM
3810 GOSUB 2480
3820 PRINT TAB(7);"-----JUEGOS/ASIGNA-----"
3830 PRINT
3840 PRINT CHR$(18);CHR$(31);"F1";CHR$(146);CHR$(144);
3850 PRINT "-MENU PRINCIPAL.";
3860 PRINT CHR$(18);CHR$(31);"F4";CHR$(146);CHR$(144);
3870 PRINT "-CAMBIA JUEGO.";
3880 PRINT CHR$(18);CHR$(31);"F2";CHR$(146);CHR$(144);
3890 PRINT "-RESUMEN.";
3900 PRINT CHR$(18);CHR$(31);"F5";CHR$(146);CHR$(144);
3910 PRINT "-COPIA DE ROM.";
3920 PRINT CHR$(18);CHR$(31);"F3";CHR$(146);CHR$(144);
3930 PRINT "-ASIGNA PARRILLA.";
3940 PRINT CHR$(18);CHR$(31);"F6";CHR$(146);CHR$(144);
3950 PRINT "-EDITA CARACTER.";
3960 PRINT
3970 GET A$

```

```

3980 IF A$="" THEN GOTO 3970
3990 LET A=ASC(A$)
4000 IF A=133 THEN GOSUB 2110:RETURN
4010 IF A=134 THEN GOSUB 4640:GOTO 3810
4020 IF A=135 THEN GOSUB 4210:GOTO 3810
4030 IF A=136 THEN GOSUB 4090:GOTO 3810
4040 IF A=137 THEN GOSUB 4460:GOTO 3810
4050 IF A=138 THEN GOSUB 4330:GOTO 3810
4060 GOTO 3970
4070 REM
4080 REM *****
4090 REM * CAMBIA JUEGO *
4100 REM *****
4110 REM
4120 PRINT "JUEGO (0-1) ? ";
4130 GET A$
4140 IF A$="" OR A$<>"1" AND A$<>"0" THEN GOTO 4130
4150 LET JU=VAL(A$)
4160 PRINT A$
4170 GOSUB 2370
4180 RETURN
4190 REM
4200 REM *****
4210 REM * ASIGNA PARRILLA *
4220 REM *****
4230 REM
4240 GOSUB 4780
4250 LET D=JU*2048+CA*8+57344!
4260 FOR G=0 TO 7
4270   POKE D+G,C(G)
4280   LET CA%(JU*256+CA,G)=C(G)
4290 NEXT G
4300 RETURN
4310 REM
4320 REM *****
4330 REM * EDITA CARACTER *
4340 REM *****
4350 REM
4360 GOSUB 4780
4370 FOR G=0 TO 7
4380   LET C(G)=CA%(JU*256+CA,G)
4390 NEXT G
4400 GOSUB 1490
4410 LET X=0
4420 LET Y=0
4430 RETURN
4440 REM
4450 REM *****
4460 REM * COPIA DE ROM *
4470 REM *****
4480 REM
4490 GOSUB 4780
4500 POKE 56334!,0
4510 POKE 1,51
4520 LET D=JU*2048+CA*8+53248!
4530 FOR G=0 TO 7
4540   LET C(G)=PEEK(D+G)
4550 NEXT G
4560 POKE 1,55
4570 POKE 56334!,1
4580 GOSUB 1490
4590 LET X=0
4600 LET Y=0
4610 RETURN
4620 REM

```

TRUCOS Y RUTINAS BASICAS

```

4640 REM * RESUMEN DE CARACTERES CON LOS QUE SE ESTA TRABAJANDO *
4650 REM *****
4660 REM
4670 POKE 56576!,196
4680 POKE 648,196
4690 POKE 53272!,PEEK(53272!) AND NOT 6 OR 8
4700 GET A$
4710 IF A$="" THEN GOTO 4700
4720 POKE 53272!,21
4730 POKE 648,4
4740 POKE 56576!,199
4750 RETURN
4760 REM
4770 REM *****
4780 REM * INPUT CHARACTER *
4790 REM *****
4800 REM
4810 PRINT "INTRODUZCA CARACTER:0-NORMAL,1-INVERSO"
4820 GET A$
4830 IF A$="" OR A$<>"0" OR A$<>"1" THEN GOTO 4820
4840 LET T1=VAL(A$)
4850 PRINT "CARACTER ? ";
4860 GET T$
4870 IF T$="" THEN GOTO 4860
4880 PRINT T$
4890 GOSUB 1800
4900 LET CA=T1*128+T
4910 GOSUB 2370
4920 RETURN

```

Aparte de todo esto, tenemos una serie de funciones que nos permiten variar el contenido de lo que tengamos en la parrilla. A estas funciones se accede pulsando las teclas de función y son:

F1 — Invierte caracter. Al pulsar esta tecla de función hacemos que todos los puntos del carácter que en estos momentos estamos editando se inviertan. Esto es, los puntos que están a cero se ponen a uno y los que están a uno se ponen a cero. Si, por ejemplo, lo que tenemos en la parrilla es la letra "O" mayúscula, tras pulsar esta tecla seguiremos viendo dicha letra, pero en vez de estar formada por puntos negros sobre un fondo sin puntos, la veremos sin puntos sobre un fondo de puntos negros.

F2 — Rotación a la derecha. Con esta segunda opción conseguimos que el contenido de la parrilla se mueva un punto a la derecha. La columna que está más a la izquierda se colocará en la posición inmediatamente siguiente a la derecha. La que ocupaba dicha posición anteriormente se pondrá en la inmediatamente contigua a la derecha. Hasta que la que se encuentra más a la derecha desaparece y reaparece a la izquierda de la parrilla.

F3 — Rotación a la izquierda. Si pulsamos la tecla de función F3 obtenemos el mismo resultado que si pulsamos la tecla F2, pero en este caso la rotación y traslación de las columnas de puntos se realizan de derecha a izquierda. La columna que está más a la izquierda desaparece y reaparece en la columna que está más a la derecha.

F4 — Espejo horizontal. Al pulsar esta tecla el contenido de la pantalla cambiará y lo veremos como si estuviésemos viendo la parrilla desde detrás de la pantalla del televisor. Los que estaban a la derecha los vamos a ver a la izquierda los que estaban a la izquierda los veremos a la derecha. Lo que vemos es la imagen que veríamos reflejada en un espejo si pusiésemos la parrilla frente a uno.

F5 — Espejo vertical. El resultado de este comando es muy parecido al anterior. En este caso lo que hacemos es darle la vuelta al contenido de la parrilla de forma que si hemos dibujado a un hombre de pie, tras pulsar la tecla F5 lo veremos cabeza abajo. Lo que está arriba se pone abajo y lo que está abajo se coloca arriba.

F6 — Save/Load. Al pulsar esta opción

nos saldrá otro pequeño menú desde donde podremos salvar en cinta o leer desde cinta el juego de caracteres que estamos definiendo o que, en otro momento, definimos.

F7 — Juegos/Asigna. Este comando nos muestra otro menú en pantalla con otra serie de opciones.

F8 — Borra parrilla. Tras pulsar la tecla F8, el contenido de la parrilla de dibujo se borrará y estará todo listo para empezar la definición de otro carácter.

Hemos visto que las teclas F6 y F7 hacen que se nos aparezcan en pantalla otros menús distintos del general. Veamos ahora qué nos aparece en cada uno de dichos menús y para qué sirve cada una de las opciones que contemplan.

Si pulsamos la tecla F6 (SAVE/LOAD), nos aparecerá un menú con las siguientes opciones:

F1 — Save juego. Con esta opción podremos grabar el juego de caracteres sobre el que estamos trabajando, en una cinta de cassette. El ordenador nos preguntará qué nombre queremos darle al fichero que va a contener las definiciones. La grabación se realiza empezando por el carácter número 0 y terminando por el 255. Se graban las ocho líneas que componen cada carácter sin separación entre ellas ni entre los caracteres. La grabación está hecha de forma que cuando el usuario necesite cargar dicho juego de caracteres lo pueda realizar con un simple bucle. Un ejemplo de cómo se pueden leer dichos caracteres es el siguiente:

```
100 OPEN 1, 1, 0, "NOMBRE DEL FICHERO"
110 FOR I = 0 TO 255
120 FOR J = 0 TO 255
130 INPUT 11, A
140 POKE DIRECCION+I*8+J, A
150 NEXT J
160 NEXT I
170 CLOSE 1
```

Hay que explicar dos cosas de este listado. La primera es que donde pone:

"NOMBRE DEL FICHERO"

cada usuario tendrá que poner el nombre del fichero que dio cuando grabó el juego de caracteres. En la línea 140 se hace mención a una variable numérica llamada DIRECCION. Esta variable tiene que ser inicializada por el usuario y tiene que tener almacenada la dirección donde queremos colocar nuestro nuevo juego de caracteres.

F2 — Load juego. Con esta opción podremos leer un juego que previamente habíamos grabado en cinta y que no hemos terminado todavía o que queremos modificar.

F3 — Retorno al menú principal. Como el mismo nombre de la operación indica, tras pulsar esta tecla de función nos volveremos al menú principal, y por tanto, seguiremos definiendo caracteres.

Si lo que pulsamos desde el menú principal fue la tecla de función F7, nos aparecerá otro menú cuyas opciones son las siguientes:

F1 — Menú principal. Nos volvemos al menú principal para continuar con la definición de caracteres.

F2 — Resumen. Pulsando esta tecla nos aparecerá por pantalla un resumen de los caracteres con los que estamos trabajando en este momento. Para volver al menú hace falta pulsar una tecla.

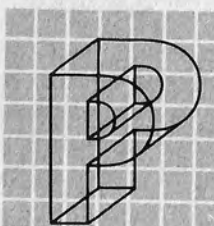
F3 — Asigna parrilla. Tras definir nuestro carácter en la parrilla, siempre tenemos que venir a esta opción. Con ella almacenamos el contenido de la parrilla en memoria. El carácter que quedará definido será el carácter que nosotros dijimos que queríamos definir antes de empezar a dibujarlo.

F4 — Cambia juego. Hemos dicho más arriba que con este programa podíamos tener dos juegos distintos de caracteres. Al dar a esta opción, el ordenador nos preguntará con cuál de los dos juegos queremos trabajar. El juego que nosotros elijamos será el que veremos y el que se grave cuando usemos la opción SAVE del menú anterior.

F5 — Copia de ROM. Al principio, cuando ejecutamos el programa, los caracteres que tenemos almacenados son una copia de los de la ROM. Puede que en algún momento de la definición necesitemos la imagen de algún carácter de la ROM para actuar sobre él y que dicho carácter ya esté borrado en nuestros dos juegos de caracteres. Pulsando esta tecla de función podemos elegir el carácter que queremos recuperar de la ROM y que queremos editar.

F6 — Edita caracter. Esta opción es muy parecida a la anterior, pero con la diferencia de que el carácter que se edita en la parrilla para su modificación pertenece al juego de caracteres con el que estamos trabajando en este momento.

■ Tarjeta de conexiones asíncronas



PARA la conexión de dispositivos asíncronos ya se han presentado los circuitos básicos y las características de los conectores que se emplean normalmente. Es necesario describir el diseño de los circuitos que se encargan de la serialización de los datos y que permiten el uso práctico de la conexión serie.

Existen numerosos circuitos integrados comerciales que realizan la mayor parte de las funciones de la conexión serie y que se encuentran fácilmente en los comercios de componentes electrónicos. También son fáciles de encontrar tarjetas completas para aquellos ordenadores personales que no disponen de forma estándar de este tipo de conexión.

■ Diagrama de bloques

Es el circuito fundamental, pues realiza la serialización y deserialización de los datos.

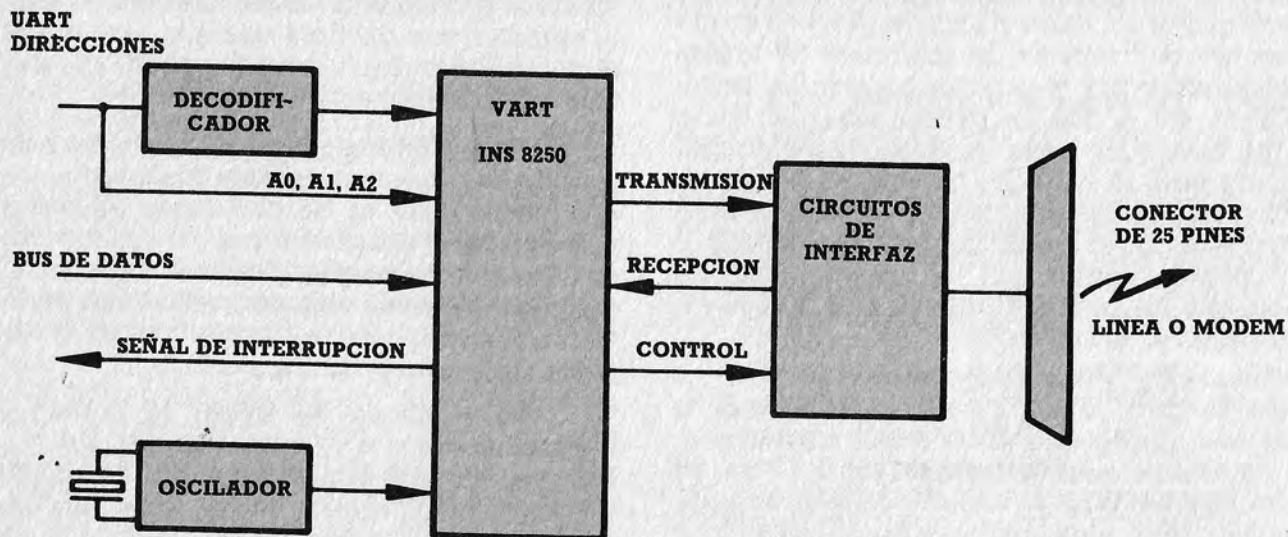


Fig. 1. Diagrama de bloques.

La tarjeta se compone de los siguientes elementos fundamentales:

- Decodificador de direcciones. Servirá para ubicar los puertos necesarios para la transmisión y recepción de datos y los bits de control.
- Interfaz con el bus de datos. Será bidireccional.
- UART. (UnAsynchronous Receive Transmit.)

Es programable en sus parámetros principales.

- Oscilador. Determina la referencia para las frecuencias a las cuales se puede transmitir.
- Interfaz externa. Adapta los niveles propios de la UART a los de los circuitos exteriores, como, por ejemplo, un MODEM.

Al igual que otros dispositivos previamente descritos, la tarjeta de conexiones asíncronas se conecta al ordenador personal a través de los buses de datos y direcciones. Del bus de direcciones se tomarán las líneas necesarias para decodificar tantas direcciones como requiera el circuito de la UART. Si la UART dispone de circuitos internos de selección, será necesario solamente decodificar la parte alta de la dirección, dejando para los bits de menor peso la conexión a las entradas del circuito. Al bus de datos se conectan los ocho hilos de datos del circuito de la UART.

El circuito que se utiliza como UART es el INS8250, que posee circuitos internos para la generación de cualquier frecuencia de bit de forma programable. Existen otros circuitos igualmente de fácil adquisición que no poseen esta circuitería y que, por tanto, requieren osciladores externos de frecuencias múltiplo de la necesaria para la transmisión. Veremos algunos de ellos más adelante, pues se les encuentra en tarjetas de conexión a los ordenadores personales.

El circuito INS8250 posee, además, otras características que le hacen muy atractivo para el diseño de una interfaz serie asíncrona, como pueden ser:

- Dispone de doble registro de almacenamiento del dato para entrada y salida, por lo que simplifica el programa de lectura/escritura y la necesidad de sincronización precisa.

- Configura automáticamente el carácter según el formato asíncrono, añadiendo los bits de arranque, parada y paridad.

- La paridad se genera automáticamente con el dato transmitido y se verifica en el dato recibido, si se ha seleccionado la opción en el registro de control.

- Se pueden controlar independientemente los orígenes de interrupción para las funciones de transmisión, recepción y señales de conexión al MODEM.

- La frecuencia de señalización de datos puede programarse mediante un divisor interno entre 1 y 2^{16} , para proporcionar la señal de reloj de 16 veces la de salida.

- Dispone de entrada independiente para el reloj de recepción, por lo que puede emplearse un solo circuito para conexiones asimétricas como es la de Videotex.

- Se pueden programar las características de la interfaz:

- 5, 6, 7 u 8 bits por cada carácter.
- Paridad par, impar, o no paridad.
- 1, 1 1/2 ó 2 bits de parada.

- Dispone de circuito detector de falso bit de arranque, eliminándose así en recepción los falsos caracteres generados por ruido.

- Generación y detección de señal de corte de línea.

- Posibilidad de ejecución de diagnósticos internos, mediante la activación de circuitos que puentean el conector. De esta forma puede verificarse el funcionamiento de todo el sistema excepto la conexión exterior.

- Posibilidad de activación de interrupciones con control de prioridad por diferentes orígenes.

El protocolo, propiamente dicho, de la comunicación debe ser realizado por el programa de control. Generalmente los sistemas operativos disponen de un soporte simplificado de control de las comunicaciones asíncronas, siendo responsabilidad del programa de aplicación la interpretación de las secuencias de datos y comandos que se reciban.

■ Direcciones internas de la UART

El circuito integrado de la UART posee internamente diversos registros que pueden escribirse o leerse de acuerdo con una dirección. El significado de cada uno de ellos es el siguiente:

- 0 = Registro de transmisión o recepción, según que la operación sea de entrada o salida. También en esta dirección se accede a la parte baja del divisor que genera la frecuencia de transmisión. Para acceder al divisor es necesario poner el bit DLAB en uno. Este bit se encuentra en la dirección del registro de control de línea, bit 7.

- 1 = Registro de permiso de interrupciones. También se accede a la parte alta del registro divisor de frecuencia.

- 2 = Registro de identificación de interrupción. Sirve para determinar el origen de la interrupción por la cual se entró en la rutina de tratamiento.

- 3 = Registro de control de línea.

- 4 = Registro de control del MODEM.

- 5 = Registro de estado de línea.

- 6 = Registro de estado del MODEM.

Las direcciones se consiguen mediante la conexión de los pines correspondientes a los bits A0, A1 y A2 del bus de direcciones. La selección general del circuito se obtiene activando las entradas CS0, CS1 y -CS2, mediante un decodificador de direcciones como el empleado para generar SELW0 o SELR0.

■ Descripción de los pines de circuito INS8250

El circuito INS8250, de cuarenta pines, dispone de entradas y salidas para todas las señales necesarias para realizar una transmisión asíncrona completamente automática. Describiremos el significado de las señales principales y sus especificaciones de interés.

Señales de **entrada**:

- Selección del chip (CS0, CS1, CS2). Pines 12, 13 y 14. Sirven para seleccionar el circuito completo. Deberán estar a nivel 1 las entradas CS0 y CS1 y en 0 la entrada CS2. Con las tres entradas puede realizarse la decodificación o utilizarse una decodificación externa, como ya se ha indicado en circuitos anteriores.

- Muestreo de datos de entrada (DISTR y -DISTR). Pines 22 y 21. Especifican el instante en el que se realiza la transferencia en entrada. Se conectan a la señal del bus relacionada con la operación de entrada. Solamente es necesario activar una de ellas, según que el sistema proporcione nivel alto o bajo en la conexión.

- Muestreo de datos de salida (DOSTR y -DOSTR). Especifican el instante en el que se realiza la transferencia en salida. Se conecta de igual manera que la anterior.

- Selección de registro (A0, A1 y A2). Pines 26, 27, y 28. Seleccionan el registro interno de acuerdo con el código generado por las tres señales y según lo especificado en las direcciones internas de la UART.

- Reset general (MR). Pin 35. Al activarse esta entrada a nivel alto se borran todos los registros, excepto el de recepción y los de divisores de frecuencia.

- Reloj de recepción (RCLK). Pin 9. La frecuencia deberá corresponder a 16 veces la de la señal de datos.

- Señal de datos serie de entrada (SIN). Pin 10. Los datos proporcionados por el MODEM o el dispositivo externo.

- Preparado para transmitir (CTS). Pin 36. Es la señal proporcionada por el MODEM indicativa de que pueden enviarse los datos serializados. Esta entrada tendrá su reflejo en el bit 4 del registro de estado del MODEM. Además se dispone de una indicación de cambio desde la última consulta en el bit 0 del mismo registro. Todo cambio en el bit CTS genera interrupción, si se ha programado la máscara para que así ocurra.

- Equipo de datos preparado (DSR). Pin 37. Indica que el equipo externo o MODEM está encendido y, por tanto, se puede operar. Los bits 5 y 1 del registro de estado del MODEM indican cómo se encuentra esta señal y si se efectuaron cambios desde la última consulta. Igualmente puede producir interrupción si la máscara se programa para ello.

- Recepción de señal de datos (DCD). Pin 38. La presencia de esta señal indica que el MODEM ha detectado portadora. Los bits 7 y 3 del registro de estado del MODEM indican cómo se encuentra la señal en ese momento y si hubo transición desde la última consulta.

- Indicador de llamada (RI). Pin 39. Es la entrada para la recepción de la señal de llamada proporcionada por el MODEM. Los bits 6 y 2 del registro de estado del MODEM tienen significado similar al de las señales anteriores.

Señales de **salida**:

- Terminal de datos preparado (DTR). Pin 33. Indica que el circuito está preparado para operar. Se programa mediante escritura en el registro de control del MODEM, bit 0.

- Petición de envío (RTS). Pin 32. Informa al MODEM que el circuito está preparado para transmitir. Se programa mediante escritura en el registro de control del MODEM, bit 1.

- Frecuencia de datos de salida (BAUDOUT). Proporciona al exterior la frecuencia generada mediante los divisores internos de acuerdo con la programación. Si se utiliza la misma frecuencia para la recepción, se cablea esta señal a la entrada RCLK.

- Datos serie de salida (SOUT). Pin 11. Es la señal de datos que alimentará el MODEM o el equipo exterior, a través de la interfaz física. En reposo se encuentra a nivel 1, correspondiente a señal de Marca.

- Interrupción (INTRPT). Pin 30. Se activa con cualquiera de las condiciones internas que hayan sido permitidas en el registro de máscara.

Señales **bidireccionales**:

— Datos (D7-D0). Pines 1 a 8. Este conjunto de señales es de tipo de tres estados, para ser conectado directamente a un bus de datos de un micro de 8 bits. Todas las transferencias de datos, estados y control se hacen a través de estos hilos.

— Reloj externo (XTAL1, XTAL2). Pines 16 y 17. Estas entradas proporcionan la referencia del oscilador. Puede ser un cristal de cuarzo o la salida de un oscilador externo.

■ **Registros de control**

Las características propias de la transmisión se especifican mediante el registro de control de línea. Los diferentes bits de este registro tienen el siguiente significado:

— 0 y 1 = Selección de longitud del carácter de acuerdo con la tabla:

- 00 = 5 bits.
- 01 = 6 bits.
- 02 = 7 bits.
- 03 = 8 bits.

— 2 = Número de bits de parada. Si es 0, se genera y detecta un solo bit. Si es uno, depende de la longitud del carácter. Para 5 bits, se generan y detectan 1 1/2 bits, para 6 o más se generan y detectan 2 bits.

— 3 = Generación de paridad. Si es 1, se genera paridad en transmisión y se comprueba en recepción.

— 4 = Paridad par. Un 1 significa paridad par, un 0 significa paridad impar.

— 5 = Selecciona el modo de transmisión de la paridad. Si el bit 3 es 1 y el bit 5 también es 1, se transmite bit de paridad y se detecta en recepción como un 0 si el bit 4 es 1 o como un 1 si el bit 4 es 0.

— 6 = Forzar señal de espacio. Sirve para producir el equivalente a un corte, indicando al terminal en el otro extremo la condición de atención. Se denomina Break en el argot habitual.

— 7 = DALB (Divisor Latch Access Bit). Es el bit de control para acceso al divisor del generador de frecuencia, que ocupa la misma dirección de puerto que los datos.

■ **Generador de frecuencia programable**

El circuito interno de generación de la señal de reloj se accede a través de las direc-

ciones de puerto 0 y 1. Para cargarlo es necesario programar el bit DALB a 1, al principio de la operación. La frecuencia recomendada para la generación de las frecuencias de reloj normalizadas es de 1,8432 MHz. Puede utilizarse cualquier otra frecuencia estable, pero entonces los valores del divisor programable deberán ser modificados en proporción. La frecuencia recomendada permite conseguir las referencias con mínimo error.

La siguiente tabla da los valores a cargar en el divisor interno para conseguir las frecuencias de transmisión normalizadas:

- 50 : X'900' (En notación hexadecimal).
- 110 : X'417'.
- 300 : X'180'.
- 600 : X'0C0'.
- 1200 : X'060'.
- 2400 : X'030'.
- 4800 : X'018'.
- 9600 : X'00C'.

Pueden conseguirse otras frecuencias, para aplicaciones especiales, y que resultan interesantes en comunicación local con otros dispositivos próximos o entre ordenadores.

■ **Registro de estado de línea**

En todo instante está disponible un registro con indicaciones del resultado de la última operación, tanto si fue transmisión como recepción. Tiene interés para conocer si es necesario tomar alguna decisión sobre la información transmitida, como, por ejemplo, repetir el mensaje o indicar que lo repita la estación contraria.

El significado de cada bit es el siguiente:

— 0 = (DR). Datos recibidos. Este bit se pone en 1 al llenarse el registro intermedio de lectura. Se borrará al leer el carácter o por programa.

— 1 = (OR). Datos de entrada sobreescritos. En el argot es Overrun, indicando que ha llegado un dato antes de que se leyera el anterior.

— 2 = (PE). Error de paridad. Indica que el carácter recibido presenta paridad diferente a la esperada, según la programación del circuito.

— 4 = (BI). Indicador de activación de ruptura en la línea, que se consigue o por ruptura física o por activación en el lado contrario del nivel de espacio por más del período de un carácter.

EL TALLER DEL HARDWARE

— 5 = (THRE). Indicador de que el registro intermedio de transmisión se encuentra vacío. Sirve para pedir que el programa presente el siguiente carácter. Puede producir interrupción si el bit de máscara correspondiente se encuentra activado.

— 6 = (TSRE). Indicador de que el registro de desplazamiento de salida se encuentra vacío.

— 7 = Siempre a 0.

■ Registro de identificación de interrupción

El origen de las interrupciones se indica al programa de control mediante los bits de este registro. Las interrupciones se encuentran jerarquizadas por orden:

— 11 = Estado de recepción de línea (permiso en Bit 2 de IER).

— 10 = Datos recibidos (permiso en Bit 0 de IER).

— 01 = Registro de salida vacío (permiso en Bit 1 de IER).

— 00 = Estado de la interfaz del MODEM (permiso en bit 3 de IER).

El registro IIR indica en el bit 0 si hay alguna interrupción activada. En los bits 1 y 2 indica el origen de la interrupción más prioritaria de acuerdo con la tabla anterior.

■ Registro de permiso de interrupciones

Este registro determina los orígenes de interrupción que están activos. Un 1 en el bit correspondiente indica que el origen de interrupción está activo. Los bits de programación están reflejados en la tabla anterior entre paréntesis. Los demás bits estarán siempre a 0.

■ Registro de control de la interfaz con el MODEM

Permite definir las condiciones de la conexión al MODEM externo o al otro equipo en modo local. Sólo utiliza los 5 bits más bajos con el siguiente significado:

— 0 = Control de terminal de datos preparado (DTR).

— 1 = Control de petición de envío (RTS).

— 2 y 3 = Control de las salidas auxiliares OUT1 y OUT2, que están disponibles para el usuario.

— 4 = Control de activación del bucle de comprobación. Permite el ensayo de la comunicación sin salida al exterior. Si se realiza una transmisión simulada, se recibirá inmediatamente, pudiéndose comprobar la circuitería interna en su totalidad.

■ Registro de estado del MODEM

El estado de las líneas de interfaz con el MODEM o el periférico se conoce a través de este registro, de acuerdo con la siguiente distribución de bits:

— 0 = Cambio en CTS. Indica que desde la última lectura se ha producido un cambio en la línea CTS.

— 1 = Cambio en DSR. Igualmente para DSR. Estas variaciones podrían indicar un fallo momentáneo de la transmisión o el comienzo de una nueva conexión.

— 2 = Activación de RI. Indica que el detector de llamada ha subido a nivel 1.

— 3 = Cambio en detección de portadora. Igual que los bits 0 y 1.

— 4 = CTS complementada.

— 5 = DSR complementada.

— 6 = RI complementada.

— 7 = DCD complementada.

■ Programa de control

Para la comunicación con los dispositivos externos es necesario realizar un programa que active el circuito de la tarjeta al comenzar la operación y que después ejecute las transferencias desde el programa y los registros de transmisión y recepción de la UART.

Los ordenadores personales generalmente disponen de soporte de un puerto como dispositivo asíncrono, pudiendo definirse en el lenguaje de soporte básico como unidades con las que se puede mantener comunicación de la misma forma que con la impresora local. Realmente han de disponer de rutinas internas para realizar las funciones de transferencia de datos y control de las operaciones individuales. Según el tipo de ordenador las ins-

trucciones a ejecutar serán extensiones del lenguaje BASIC o llamadas a rutinas en lenguaje de máquina que pueden encontrarse en la memoria RAM normal o en una ROM incluida en la tarjeta del adaptador. En cualquier caso será necesario contar con un programa que llene el registro de salida carácter a carácter y que tome del registro de entrada los datos que lleguen y los ensamble en un vector.

El programa que realice la comunicación puede ser síncrono con la operación, es decir, lleve el control de cada carácter por observación de los bits de estado internos de la UART. Es una forma sencilla de soporte, pero tiene el inconveniente de ocupar todo el tiem-

po de la unidad central y, por tanto, no permite hacer otra cosa. Si, por el contrario, solamente se dialoga con la UART cada vez que ésta reclama servicio a través de las interrupciones, el ordenador podrá realizar otras funciones, mientras la UART ensambla y desensambla los bits. Este modo de programación es algo más delicado, pero es el que normalmente se emplea, por ser más eficiente.

El programa siguiente, en el ordenador IBM-PC define las condiciones de comunicación y realiza las funciones de un terminal sencillo, conectado a otro igual, para intercambiar mensajes:

```

10 REM SIMULACION DE TERMINAL.
15 REM PARA IBM-PC
20 REM PROGRAMA DE COMUNICACION ASINCRONA
30 REM PARA INTERCAMBIO ENTRE DOS ORDENADORES
40 REM LA COMUNICACION ES SIMULTANEA.
70 REM CON CABLEADO DE CONEXION A TRES HILOS
80 CR$=CHR$(13)
90 CLS : KEY OFF
100 PRINT "COMUNICACION ENTRE ORDENADORES"
110 LOCATE 3,1:PRINT "TRANSMISION"
120 LOCATE 14,1:PRINT "RECEPCION"
130 LINEAT=4 : LINEAR=16
140 REM ACTIVACION DEL PROGRAMA DE INTERRUPCION
150 ON COM(1) GOSUB 230
160 OPEN "COM1:9600,E,7,1,LF" AS #1
170 COM(1) ON
180 N=0: M=0
190 PRINT #1,"COMIENZO DE LA TRANSMISION ..."
200 LOCATE 4,1:PRINT "_";
205 REM BUCLE PRINCIPAL DE PRESENTACION
210 T$=INKEY$
220 IF T$="" THEN 210 ELSE N=N+1
230 LOCATE LINEAT,N
240 PRINT T$;" ";
250 PRINT (#1,T$);
260 IF T$=CR$ THEN LOCATE LINEAT,N:PRINT " ";GOSUB 360: N=0
270 GOTO 210
280 REM LECTURA DE CARACTERES RECIBIDOS
290 D$=INPUT$(1,#1)
300 IF D$=CR$ THEN GOSUB 400: RETURN
310 M=M+1
320 LOCATE LINEAR,M
330 PRINT D$;
340 IF D$=CR$ THEN LINEAR=LINEAR+1
350 RETURN
360 REM INCREMENTO DE LINEA TRANSMITIDA
370 LINEAT=LINEAT+1
380 IF LINEAT>12 THEN LINEAT=4
390 RETURN
400 REM INCREMENTO DE LINEA RECIBIDA
410 LINEAR=LINEAR+1
420 IF LINEAR>24 THEN LINEAR=16
430 RETURN

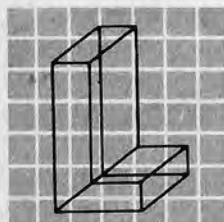
```

El programa activa las rutinas internas de comunicación mediante la sentencia OPEN, indicando la velocidad, número de bits de parada y la paridad. Además se indican los tiem-

pos de espera para las señales de la interfaz con el MODEM. La activación de la máscara de interrupción se realiza exactamente en el momento de ejecutar COM(1) ON.

NATURALEZA Y TECNOLOGIA

■ Cambios de estado



AS partículas que forman los cuerpos se mantienen unidas debido a una serie de fuerzas que actúan entre ellas. Al dar calor a una sustancia, el movimiento de sus partículas aumenta, esto hace que las distancias entre estas partículas aumenten, ya que este movimiento hace decrecer la intensidad de las fuerzas que las mantienen unidas.

Si la cantidad de calor dada y, por tanto, el incremento de energía térmica del cuerpo es muy grande, las fuerzas de unión entre sus partículas se rompen en parte, con lo que el cuerpo pasa a un estado nuevo, produciéndose el fenómeno conocido como *cambio de estado*.

Casi todos los cambios de estado se rigen por las siguientes dos leyes:

- Las transformaciones en igualdad de condiciones siempre se producen a igual temperatura.
- Mientras se está realizando la transformación, no hay variación de la temperatura.

Pueden definirse entonces tres tipos de presentación para los cuerpos: sólidos, líquidos y gases. La manera de distinguir los tres estados es a través de las propiedades de su forma, volumen y fluidez.

— Las transformaciones en igualdad de condiciones siempre se producen a igual temperatura.

— Mientras se está realizando la transformación, no hay variación de la temperatura.

Pueden definirse entonces tres tipos de presentación para los cuerpos: sólidos, líquidos y gases. La manera de distinguir los tres estados es a través de las propiedades de su forma, volumen y fluidez.



Fig. 1. Cuadro resumen de los cambios de estado.

Sólido es el único estado que todas las sustancias pueden adoptar, pues antes de fun-

dir o vaporizarse algunas se descomponen. Es el estado más ordenado y se da cuando las energías de enlace entre las partículas son mayores que la agitación interna (energía térmica). Si le damos calor, el sólido comienza a dilatarse (el aumento de energía térmica no es muy grande), separándose un poco las partículas; si se sigue dando calor las partículas adquieren la libertad suficiente para fluir y ser líquido. A medida que la agitación interna es mayor, las distancias medias entre partículas aumentan y se debilitan las fuerzas de unión. Cuando la energía de agitación interna es mucho mayor que la de unión entre las partículas, éstas se liberan del líquido pasando a estado gas.

■ Calor de cambio de estado

Cualquier sustancia que pase por los tres estados sigue un proceso similar al que sufre el hielo y que se describe en el cuadro siguiente:

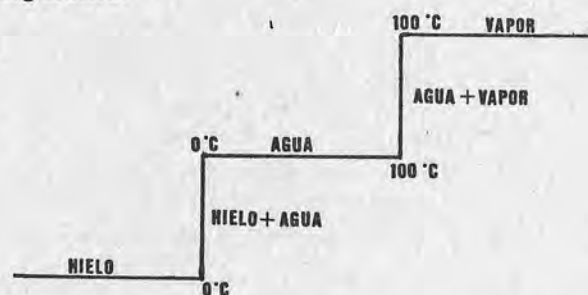


Fig. 2. Cambios de estado en el agua (hielo a 0° C a agua y vapor a 100° C).

Para pasar de sólido a líquido y viceversa hay que dar una cantidad de calor o quitar la misma cantidad de calor. Estas cantidades son constantes en igualdad de condiciones. Si se quiere pasar el hielo a agua a 0° C habrá que dar por cada gramo de hielo 80 calorías, obteniéndose el agua a esa temperatura. Es más, a 0° C, el hielo y el agua coexisten, sin que el hielo se funda ni el agua pase a hielo, teniendo que dar o quitar calor para pasar de un estado a otro o viceversa.

Algo parecido ocurre con el agua y el vapor de agua. Ambos estados pueden coexistir, suponiendo unas condiciones normales, a 100° C de temperatura; teniendo que dar 540

calorías por gramo si se quiere pasar un gramo de agua a vapor, o quitar 540 calorías por gramo en caso contrario.

Ambos calores son los llamados calores de fusión y evaporación.

Calor de fusión de una sustancia es el calor necesario para fundir un gramo de ella (agua: 80 cal/g).

Calor de evaporación de una sustancia es el calor necesario para vaporizar un gramo de sustancia (agua: 540 cal/g).

Es necesario conocer el calor a utilizar para pasar de una temperatura a otra cuando

continúa en un mismo estado. La fórmula que nos dice ese calor cedido/ganado es:

$$Q = m.c. (t_2 - t_1)$$

donde:

Q: Calor cedido/ganado por la sustancia (calorías).

m: Masa en gramos de la sustancia (g).

c: Calor específico de la sustancia (cal/g°C).

t₂: Temperatura final de la sustancia (°C).

t₁: Temperatura inicial de la sustancia (°C).

```

10 REM *****
15 REM *
20 REM * CAMBIOS DE ESTADO *
25 REM * *
30 REM * APLICACION EN EL AGUA *
35 REM * *
40 REM *****
45 CLS
46 PRINT TAB(7); "EL AGUA: CAMBIOS DE ESTADO."
48 PRINT TAB(7); "-----"
50 PRINT
80 PRINT TAB(2); "EL AGUA PASARA POR LOS TRES ES-"
85 PRINT "TADOS: SOLIDO (HIELO), LIQUIDO,"
90 PRINT "Y GAS (VAPOR DE AGUA). "
92 PRINT
95 PRINT TAB(2); "PARA LLEGAR A CADA UNO DE ELLOS"
100 PRINT "ES NECESARIO APORTAR UN CALOR QUE"
105 PRINT "PERMITA EL CAMBIO."
110 PRINT
115 PRINT TAB(2); "ASI, PARA PASAR DE HIELO A CERO"
120 PRINT "GRADOS A AGUA A CERO GRADOS, ES"
125 PRINT "NECESARIO APORTAR 80 CALORIAS"
130 PRINT "POR CADA GRAMO DE HIELO. Y PARA"
140 PRINT "PASAR DE AGUA A 100 GRADOS A VA"
150 PRINT "POR SON NECESARIAS 540 CALORIAS."
151 PRINT
152 PRINT TAB(2); "EL PROGRAMA VA A BASARSE EN LOS"
153 PRINT "CAMBIOS DE ESTADO QUE TIENEN LU"
154 PRINT "GAR EN EL AGUA."
158 FOR I=1 TO 500:NEXT I
160 PRINT:INPUT "(PULSA UNA TECLA PARA CONTINUAR)";P$
170 GOSUB 3000
180 PRINT:PRINT
190 PRINT TAB(3); "PUEDES LLEGAR A DETERMINAR LA"
195 PRINT "CANTIDAD DE CALOR A APORTAR PARA"
200 PRINT "MODIFICAR EL ESTADO DEL AGUA"DELETE 165
210 PRINT "TAL QUE:"
220 PRINT
230 PRINT TAB(3); "1. PASAR HIELO DE -T GRADOS A"
235 PRINT TAB(6); "AGUA A CERO GRADOS."
238 PRINT
240 PRINT TAB(3); "2. PASAR HIELO DE -T GRADOS A"
250 PRINT TAB(6); "AGUA A T GRADOS."
260 PRINT
270 PRINT TAB(3); "3. PASAR VAPOR DE 100 GRADOS A"
280 PRINT TAB(6); "AGUA A T GRADOS."
290 PRINT
300 PRINT TAB(3); "4. MEZCLA DE HIELO Y AGUA PARA"
310 PRINT TAB(6); "ALCANZAR EL CONJUNTO T GRADOS"
320 PRINT:PRINT:PRINT
322 FOR I=1 TO 400:NEXT I
330 PRINT "PULSA LA OPCION DESEADA (1-2-3-4):";
335 REM **** ELECCION DE LA OPCION ****
345 INPUT A
350 ON A GOTO 500,800,1000,1200
360 CLS
370 REM **** FIN DE PROGRAMA ****
380 PRINT:PRINT:PRINT:PRINT
390 PRINT "DESEAS VOLVER AL MENU DE OPCIO"
400 PRINT "NES (S/N)";

```

```

415 INPUT K$
420 IF K$="S" OR K$="s" THEN CLS:GOTO 180
430 IF K$("<"N" AND K$("<"n" THEN CLS:GOTO 380
440 CLS
450 LOCATE 12,10: PRINT "!!! ADIOS !!!"
460 FOR I=1 TO 5000:NEXT I: CLS
470 END
500 REM *****
502 REM * RESOLUCION DE LOS VALORES *
504 REM * PARA LA PRIMERA OPCION *
506 REM *****
510 CLS
520 GOSUB 3000
530 PRINT:PRINT:PRINT :PRINT
535 PRINT TAB(3);"CALOR PEDIDO: QT=Q1+Q2"
540 FOR I=1 TO 3000:NEXT I
550 GOSUB 2000
560 GOSUB 2300
570 CLS
580 PRINT:PRINT:PRINT :PRINT
590 PRINT TAB(7);"Y EL CALOR RESULTANTE <QT> ES:"
600 PRINT:PRINT
610 PRINT TAB(5);"QT=";Q1+Q2;"CALORIAS."
620 FOR I=1 TO 4000:NEXT I
630 GOTO 360
800 REM *****
802 REM * RESOLUCION DE LOS VALORES *
804 REM * PARA LA SEGUNDA OPCION *
806 REM *****
810 CLS
820 GOSUB 3000
830 PRINT:PRINT:PRINT :PRINT
840 PRINT TAB(6);"CALOR PEDIDO: QT=Q1+Q2+Q3."
850 FOR I=1 TO 4000: NEXT I
860 GOSUB 2000
870 GOSUB 2300
880 GOSUB 2600
890 CLS
900 PRINT :PRINT :PRINT :PRINT
910 PRINT TAB(4);"EL CALOR NECESARIO TOTAL ES:"
920 PRINT :PRINT
930 PRINT TAB(5);"QT=";Q1+Q2+Q3;"CALORIAS."
940 FOR I=1 TO 4000:NEXT I
950 GOTO 360
1000 REM *****
1002 REM * RESOLUCION DE LOS VALORES *
1004 REM * PARA LA TERCERA OPCION *
1006 REM *****
1010 CLS
1020 GOSUB 3000
1030 PRINT :PRINT :PRINT :PRINT
1040 PRINT TAB(6);"CALOR PEDIDO: QT=Q4+Q3"
1050 FOR I=1 TO 1100:NEXT I
1060 PRINT :PRINT
1070 PRINT "Q4: PARA PASAR VAPOR A 100 GRA"
1080 PRINT TAB(5);"DOS A AGUA A 100 GRADOS."
1090 FOR I=1 TO 500: NEXT I
1100 PRINT :PRINT
1120 PRINT "Q3: PARA PASAR AGUA A 100 GRA"
1130 PRINT TAB(5);"DOS A AGUA A T GRADOS."
1132 FOR I=1 TO 500: NEXT I
1136 PRINT :PRINT :PRINT
1138 PRINT :PRINT :PRINT "(ESPERA UN MOMENTO, POR FAVOR)"
1140 FOR I=1 TO 10000: NEXT I
1150 GOSUB 2800
1160 GOSUB 2600
1170 CLS
1175 PRINT :PRINT :PRINT :PRINT
1180 PRINT TAB(7);"Y EL CALOR RESULTANTE <T> ES:"
1185 PRINT :PRINT
1188 PRINT :PRINT TAB(5);"QT=";Q4+Q3;"CALORIAS."
1190 FOR I=1 TO 4000:NEXT I
1195 GOTO 360
1200 REM *****
1202 REM * RESOLUCION DE LOS VALORES *
1204 REM * PARA LA CUARTA OPCION *
1206 REM *****
1210 CLS
1220 GOSUB 3000
1230 PRINT
1240 PRINT TAB(3);"LA TEMPERATURA DEL CONJUNTO ES"
1250 PRINT "CERO GRADOS, YA QUE ES LA UNICA"

```

```

1260 PRINT "TEMPERATURA A LA QUE COEXISTEN"
1270 PRINT "EL HIELO Y EL AGUA."
1280 PRINT
1290 PRINT TAB(3); "EL CALOR PARA CONSEGUIR ESTAR"
1300 PRINT "EL CONJUNTO A <T> GRADOS ES:"
1310 PRINT :PRINT TAB(4); "QT=Q2+Q(AGUA)+Q(HIELO)"
1320 PRINT :PRINT "Q2: PASAR HIELO CERO GRADOS A"
1330 PRINT "AGUA A CERO GRADOS."
1340 PRINT :PRINT "Q(HIELO): PASAR AGUA (QUE ANTES"
1350 PRINT "ERA HIELO) DE CERO GRADOS A AGUA"
1360 PRINT "A <T> GRADOS."
1370 PRINT :PRINT "Q(AGUA): PASAR AGUA DE CERO GRA"
1380 PRINT "DOS AGUA A <T> GRADOS."
1387 FOR I=1 TO 2000:NEXT I
1390 PRINT :PRINT :PRINT "(PULSA UNA TECLA PARA CONTINUAR)";:INPUT P$
1400 GOSUB 2300
1410 GOSUB 2600
1415 LET S=Q3
1420 GOSUB 2600
1425 LET S=S+Q3
1430 CLS
1440 PRINT :PRINT :PRINT :PRINT
1450 PRINT TAB(7); "Y EL CALOR RESULTANTE <QT> ES:"
1455 PRINT :PRINT
1460 PRINT TAB(5); "QT=";Q2+S;"CALORIAS."
1470 FOR I=1 TO 3000:NEXT I
1480 GOTO 360
2000 REM *****
2002 REM * SUBROUTINA PARA Q1 (PASAR *
2004 REM * HIELO A CERO GRADOS) *
2006 REM *****
2010 CLS
2020 LET C1=.5
2030 PRINT "Q1: CALOR NECESARIO PARA PASAR"
2035 PRINT "HIELO DE -T GRADOS A CERO GRADOS"
2040 PRINT:PRINT TAB(9); "Q=m.c.(t2-t1)"
2050 PRINT:PRINT TAB(3); "DONDE:"
2060 PRINT:PRINT "Q: CALOR GANADO/CEDIDO"
2070 PRINT "m: MASA DEL CUERPO"
2080 PRINT "c: CALOR ESPECIFICO DEL CUERPO"
2090 PRINT "t2: TEMPERATURA FINAL"
2100 PRINT "t1: TEMPERATURA INICIAL"
2110 FOR I=1 TO 2000:NEXT I
2120 PRINT:INPUT "MASA DEL HIELO EN GRAMOS:";M
2130 PRINT:PRINT "c(DEL HIELO)= 0.5 cal/g.°C"
2140 PRINT:INPUT "TEMPERATURA INICIAL HIELO:";T1
2150 FOR I=1 TO 500:NEXT I
2170 LET T1=ABS(T1): LET Q1=M*C1*T1
2180 PRINT
2190 PRINT TAB(4); "Q1=";Q1;"CALORIAS"
2192 FOR I=1 TO 1500:NEXT I
2195 PRINT:PRINT:PRINT "(PULSA UNA TECLA PARA CONTINUAR)";:INPUT P$
2200 RETURN
2300 REM *****
2302 REM * SUBROUTINA PARA Q2 (PASAR *
2304 REM * HIELO DE CERO GRADOS A *
2306 REM * AGUA A CERO GRADOS) *
2308 REM *****
2310 CLS
2320 LET L=80
2330 PRINT:PRINT:PRINT "Q2: CALOR NECESARIO PARA PASAR"
2335 PRINT "HIELO DE CERO GRADOS A AGUA A"
2340 PRINT "CERO GRADOS."
2350 PRINT:PRINT TAB(9); "Q= m.L"
2360 PRINT:PRINT TAB(4); "DONDE:"
2370 PRINT:PRINT "Q: CALOR GANADO/CEDIDO"
2380 PRINT "m: MASA DEL CUERPO"
2390 PRINT "L: CALOR FUSION (80 cal/g)"
2400 PRINT:PRINT:PRINT
2410 INPUT "MASA DEL HIELO EN GRAMOS:"; M
2415 LET Q2=M*L
2420 PRINT:PRINT
2430 PRINT TAB(3); "Q2=";Q2;"CALORIAS"
2432 FOR I=1 TO 1500:NEXT I
2435 PRINT:PRINT:PRINT "(PULSA UNA TECLA PARA CONTINUAR)";:INPUT P$
2440 RETURN
2600 REM *****
2602 REM * SUBROUTINA PARA Q3 (PASAR *
2604 REM * AGUA DE CERO/CIEN GRADOS *
2606 REM * A AGUA A <T> GRADOS) *
2608 REM *****
2610 CLS

```

```

2620 LET C2=1
2630 PRINT "Q3: CALOR NECESARIO PARA PASAR"
2635 PRINT "AGUA CERO/CIEN GRADOS A T GRADOS"
2640 PRINT:PRINT TAB(9); "Q=m.c.(t2-t1)"
2650 PRINT:PRINT TAB(3); "DONDE:"
2660 PRINT:PRINT "Q: CALOR GANADO/CEDIDO"
2670 PRINT "m: MASA DEL CUERPO"
2680 PRINT "c: CALOR ESPECIFICO DEL CUERPO"
2690 PRINT "t2: TEMPERATURA FINAL"
2695 PRINT "T1: TEMPERATURA INICIAL"
2700 FOR I=1 TO 1500:NEXT I
2710 PRINT:INPUT "MASA DEL AGUA EN GRAMOS: ";M
2720 PRINT:PRINT "c (DEL AGUA)= 1 cal/g.oC"
2725 PRINT:INPUT "TEMPERATURA FINAL DEL AGUA: ";T2
2730 PRINT:INPUT "TEMPERATURA INICIAL DEL AGUA: ";T1
2735 IF T1=0 THEN LET Q3=M*C2*T2: GOTO 2760
2740 IF T1<>100 THEN CLS:GOTO 2730
2750 LET Q3=M*C2*(100-T2)
2760 PRINT
2770 PRINT TAB(4); "Q3=";Q3;"CALORIAS"
2772 FOR I=1 TO 1000:NEXT I
2775 PRINT:PRINT "(PULSA UNA TECLA PARA CONTINUAR)";:INPUT P$
2780 RETURN
2800 REM *****
2802 REM * SUBROUTINA PARA Q4 (PASAR *
2804 REM * VAPOR DE 100 GRADOS A *
2806 REM * AGUA A 100 GRADOS) *
2808 REM *****
2810 CLS
2820 LET V=540
2830 PRINT:PRINT:PRINT "Q4: CALOR NECESARIO PARA PASAR"
2835 PRINT "VAPOR A 100 GRADOS A AGUA A 100"
2840 PRINT "GRADOS."
2850 PRINT:PRINT TAB(9); "Q=m.V"
2860 PRINT:PRINT "DONDE:"
2870 PRINT:PRINT "Q: CALOR CEDIDO/GANADO"
2880 PRINT "m: MASA DEL VAPOR"
2890 PRINT "V: CALOR VAPORIZACION"
2895 PRINT TAB(5); "(540 cal/g)"
2900 PRINT:PRINT
2905 FOR I=1 TO 1500:NEXT I
2910 INPUT "MASA DEL VAPOR EN GRAMOS: ";M
2920 LET Q4=M*V
2930 PRINT:PRINT
2940 PRINT TAB(3); "Q4=";Q4;"CALORIAS."
2942 FOR I=1 TO 1000:NEXT I
2945 PRINT:PRINT "(PULSA UNA TECLA PARA CONTINUAR)";:INPUT P$
2950 RETURN
3000 REM *****
3002 REM * SUBROUTINA PARA EL DIBUJO *
3004 REM * DE LOS CAMBIOS DE ESTADO *
3006 REM * CON RESPECTO DEL AGUA *
3008 REM *****
3010 SCREEN 1
3012 CLS
3020 PRINT TAB(4); "CAMBIOS DE ESTADO EN EL AGUA"
3030 LINE (5,120)-(110,120):LINE -(110,80)
3040 LINE -(210,80):LINE -(210,40)
3050 LINE -(315,40)
3070 PRINT:PRINT
3080 PRINT TAB(26); "VAPOR"
3090 PRINT TAB(27); "100 C"
3100 PRINT:PRINT TAB(26); "AGUA+VAPOR"
3105 PRINT TAB(26); "(540 cal/g)"
3110 LOCATE 9,13:PRINT "AGUA"
3120 PRINT:PRINT:LOCATE 10,13:PRINT "0 C"
3130 LOCATE 11,28:PRINT "100 C"
3140 LOCATE 12,28:PRINT "AGUA"
3150 PRINT TAB(14); "HIELO+AGUA"
3155 PRINT TAB(14); "(80 cal/g)"
3160 PRINT:PRINT
3170 PRINT TAB(14); "0 C"
3180 PRINT TAB(14); "HIELO"
3190 PRINT
3200 PRINT "CALOR FUSION DEL HIELO: 80 cal/g"
3210 PRINT "CALOR VAPORIZACION: 540 cal/g"
3215 FOR I=1 TO 1000:NEXT I
3220 PRINT:PRINT:INPUT "(PULSA UNA TECLA PARA CONTINUAR)";P$
3230 CLS
3238 SCREEN 0
3240 RETURN

```

Modificaciones para otros equipos

ZX-SPECTRUM:

```
315 CLS
350 IF A=1 THEN GOTO 500
352 IF A=2 THEN GOTO 800
354 IF A=3 THEN GOTO 1000
356 IF A=4 THEN GOTO 1200
358 GOTO 315
470 GOTO 9999
3010 (NO SE PONE)
3030 PLOT 3,65: DRAW 73,0
3035 PLOT 77,65: DRAW 0,40
3040 PLOT 77,106: DRAW 80,0
3045 PLOT 158,106: DRAW 0,40
3050 PLOT 158,146: DRAW 80,0
3070 PRINT TAB 19;"VAPOR"
3090 PRINT TAB 19;"100 ° C"
3100 PRINT AT 6,19;"AGUA+VAPOR"
3110 PRINT AT 9,20;"100 ° C"
3120 PRINT AT 10,20;"AGUA"
3124 PRINT AT 6,9;"AGUA"
3126 PRINT AT 7,9;"0 ° C"
3130 PRINT AT 11,9;"HIELO+AGUA"
3140 PRINT AT 14,10;"0 ° C"
3150 PRINT AT 15,10;"HIELO"
3170 PRINT:PRINT
3180 PRINT "CALOR FUSION DEL HIELO: 80
CAL/G"
3190 PRINT "CALOR VAPORIZACION: 540 CAL/G"
3220 PRINT: INPUT "(PULSA UNA TECLA PARA
CONTINUAR)";P$
```

COMMODORE:

```
45 PRINT CHR$(147)
170 (NO SE PONE)
175 PRINT CHR$(147)
360 PRINT CHR$(147)
420 IF K$="S" OR K$="s" THEN PRINT
CHR$(147):GOTO 180
430 IF K$<>"N" AND K$<>"n" THEN CHR$(147):
GOTO 380
440 PRINT CHR$(147)
450 PRINT TAB(6);" ¡¡¡ ADIOS !!! "
460 FOR I=1 TO 5000: NEXT I: PRINT CHR$(147)
510 PRINT CHR$(147)
520 (NO SE PONE)
525 PRINT CHR$(147)
570 PRINT CHR$(147)
810 PRINT CHR$(147)
820 (NO SE PONE)
825 PRINT CHR$(147)
890 PRINT CHR$(147)
1010 PRINT CHR$(147)
1020 (NO SE PONE)
1025 PRINT CHR$(147)
1170 PRINT CHR$(147)
1210 PRINT CHR$(147)
1220 (NO SE PONE)
1225 PRINT CHR$(147)
1430 PRINT CHR$(147)
2010 PRINT CHR$(147)
2310 PRINT CHR$(147)
2610 PRINT CHR$(147)
2810 PRINT CHR$(147)
3000-3240 (NO SE PONEN)
```

AMSTRAD:

```
450 LOCATE 10,12: PRINT "¡¡¡ADIOS!!!"
3010 MODE 1
3030 PLOT 3,120: DRAW 73,120
3035 PLOT 73,120: DRAW 73,80
3040 PLOT 73,80: DRAW 146,80
3045 PLOT 146,80: DRAW 146,40
3050 PLOT 146,40: DRAW 219,40
3238 MODE 0
```

MATEMATICAS

■ El plano en el espacio afín. Formas de la ecuación del plano

La estructura del espacio afín real es la misma que la del plano afín. La única diferencia es la *dimensión*, que en el caso del espacio es $\dim V(R):3$.

Se pueden determinar las distintas formas de la ecuación de un plano en el espacio según:

- Conociendo un punto P y dos vectores directores (u, v) .
- Conociendo tres puntos (P, Q, R) .

1. Formas de la ecuación del plano

Podemos definir un plano como:

$$\pi \equiv \{x \in E \text{ (conjunto de puntos)} / AX = tu + sv\}$$

Sabiendo que al estar definido sobre el espacio afín se va a trabajar con tres coordenadas, entonces:

Componentes punto P : $P(P_1, P_2, P_3)$

Componentes punto Q : $Q(Q_1, Q_2, Q_3)$

Componentes punto R : $R(R_1, R_2, R_3)$

Componentes vector director u :
 $u(U_1, U_2, U_3)$

Componentes vector director v :
 $v(V_1, V_2, V_3)$

1.1. Formas de la ecuación con un punto P y dos vectores directores

— Ecuación vectorial

$$(X, Y, Z) = (P_1, P_2, P_3) + t(U_1, U_2, U_3) + s(V_1, V_2, V_3) \quad [1]$$

— Ecuaciones paramétricas

$$\begin{aligned} X &= P_1 + t.U_1 + s.V_1 \\ Y &= P_2 + t.U_2 + s.V_2 \\ Z &= P_3 + t.U_3 + s.V_3 \end{aligned} \quad [2]$$

— Ecuación general

Eliminando los parámetros " s " y " t " se obtiene el determinante y queda:

$$A \cdot x + B \cdot y + C \cdot z + D = 0 \quad [3]$$

APRENDER CON EL ORDENADOR

1.2. Conocidos tres puntos

Al tener tres puntos operando con dos de ellos obtengo dos vectores directores y en-

tonces la deducción de las ecuaciones del plano se reducen al primer caso (ver apdo. 1.1).

```
10 REM *****
15 REM *
20 REM *      ESPACIO AFIN      *
25 REM *
30 REM * ECUACIONES DE UN PLANO *
35 REM *
40 REM *****
45 CLS
50 PRINT TAB(8);"ECUACIONES DEL PLANO"
55 PRINT TAB(8);"-----"
60 PRINT:PRINT
70 PRINT TAB(5);"PUEDES LLEGAR A DETERMINAR"
80 PRINT "LAS FORMAS (PARAMETRICA Y GENE"
85 PRINT "RAL) DE LA ECUACION DE UN PLANO"
90 PRINT "EN EL ESPACIO AFIN SEGUN:"
100 PRINT
130 PRINT TAB(3);"1. CONOCIDOS UN PUNTO <P> Y"
135 PRINT TAB(6);"DOS VECTORES U Y V NO NULOS"
140 PRINT TAB(6);"QUE NO SEAN PROPORCIONALES."
145 PRINT
150 PRINT TAB(3);"2. CONOCIDOS 3 PUNTOS P,Q Y R"
170 PRINT TAB(6);"NO ALINEADOS."
175 PRINT
180 PRINT TAB(3);"3. CONOCIDOS UNA RECTA <R> Y"
190 PRINT TAB(6);"UN PUNTO EXTERIOR A ELLA."
195 PRINT:PRINT
200 PRINT:PRINT "PULSA LA OPCION DESEADA (1-2-3):";
210 INPUT A
220 ON A GOTO 320,500,800
240 CLS
242 REM *****
244 REM *      EDICION DE FINAL DE      *
246 REM *      PROGRAMA      *
248 REM *****
250 PRINT:PRINT:PRINT:PRINT
260 PRINT "DESEAS DETERMINAR LAS ECUACIONES"
265 PRINT "DE OTRO PLANO (S/N)";
268 INPUT A$
270 IF A$="S" OR A$="s" THEN GOTO 45
280 IF A$<>"N" AND A$<>"n" THEN GOTO 260
285 CLS
290 PRINT:PRINT:PRINT:PRINT
300 PRINT TAB(8);"FIN DEL PROGRAMA "
310 FOR I=1 TO 2000: NEXT I:CLS:END
320 REM *****
322 REM *      INTRODUCCION COORDENADAS *
330 REM *      DEL PRIMER CASO      *
340 REM *****
345 CLS
348 PRINT:PRINT:PRINT:PRINT
350 PRINT TAB(4);"ES NECESARIO QUE INTRODUZCAS LAS"
360 PRINT "COORDENADAS DEL PUNTO P(P1,P2,P3) "
370 PRINT "Y LAS DE LOS VECTORES NO NULOS"
380 PRINT "U(U1,U2,U3) Y V(V1,V2,V3)."
390 PRINT:PRINT:PRINT
410 INPUT "COORDENADAS PUNTO P(P1,P2,P3):";P1,P2,P3
415 PRINT
420 INPUT "COORDENADAS VECTOR U(U1,U2,U3):";U1,U2,U3
425 PRINT
430 INPUT "COORDENADAS VECTOR V(V1,V2,V3):";V1,V2,V3
440 REM *****
442 REM * COMPROBACION DE QUE LOS *
444 REM * VECTORES U Y V NO SON *
446 REM * PROPORCIONALES *
448 REM *****
450 CLS
455 PRINT:PRINT
460 IF U1*V2=V1*U2 AND U2*V3=V2*U3 THEN PRINT "LOS VECTORES <U> Y <V> SON PROPOR"
470 GOTO 1100
500 REM *****
502 REM *      COORDENADAS DE LA *
504 REM *      SEGUNDA OPCION      *
510 REM *****
```

```

515 CLS
520 PRINT:PRINT:PRINT
525 PRINT TAB(3); "LOS TRES PUNTOS P,Q Y R NO DEBEN ESTAR"
530 PRINT "ALINEADOS."
540 PRINT TAB(3); "INTRODUCE LAS COORDENADAS:"
560 PRINT:PRINT:PRINT
570 INPUT "COORDENADAS PUNTO P (P1,P2,P3) "; P1,P2,P3
575 PRINT:PRINT
580 INPUT "COORDENADAS PUNTO Q (Q1,Q2,Q3) "; Q1,Q2,Q3
585 PRINT:PRINT
590 INPUT "COORDENADAS PUNTO R (R1,R2,R3) "; R1,R2,R3
600 REM *****
610 REM * COMPROBACION DE QUE P,Q,R *
620 REM * NO ESTAN ALINEADOS *
625 REM *****
628 CLS
630 LET U1=P2-P1
640 LET U2=Q2-Q1
650 LET U3=R2-R1
660 LET V1=P3-P1
670 LET V2=Q3-Q1
680 LET V3=R3-R1
700 IF U1*V2=U2*V1 AND U2*V3=V2*U3 THEN PRINT "LOS PUNTOS P,Q Y R ESTAN ALINEADO"
S, NO DETERMINAN UN PLANO": FOR I=1 TO 2000:NEXT I:CLS:GOTO 250
710 GOTO 1100
800 REM *****
802 REM * INTRODUCCION COORDENADAS *
806 REM * TERCERA OPCION *
808 REM *****
810 CLS
820 PRINT:PRINT
830 PRINT TAB(3); "LA RECTA <R> VA A VENIR DEFINIDA"
840 PRINT "POR UN PUNTO Q Y UN VECTOR DIRECTOR U."
850 PRINT:PRINT
860 PRINT TAB(3); "INTRODUCE LAS COORDENADAS DEL PUNTO"
870 PRINT "Q Y LAS DEL VECTOR DIRECTOR V"
880 PRINT "QUE DEFINEN LA RECTA, JUNTO CON"
890 PRINT "LAS DE OTRO PUNTO P EXTERIOR A ELLA"
900 PRINT:PRINT
910 PRINT "COORDENADAS DEL VECTOR DIRECTOR"
920 PRINT "U(U1,U2,U3) DE LA RECTA: ";
930 INPUT U1,U2,U3
935 PRINT:PRINT
940 PRINT "COORDENADAS DEL PUNTO Q(Q1,Q2,Q3) QUE:";
945 PRINT "PERTENECE A LA RECTA <R>: ";
950 INPUT Q1,Q2,Q3
955 PRINT:PRINT
960 PRINT "COORDENADAS DEL PUNTO EXTERIOR P"
965 PRINT "A LA RECTA, P(P1,P2,P3): ";
968 INPUT P1,P2,P3
970 IF (Q1-P1)*U2=(Q2-P2)*U1 AND (Q2-P2)*U3=(Q3-P3)*U2 THEN GOTO 1000
980 LET V1=Q1-P1
985 LET V2=Q2-P2
990 LET V3=Q3-P3
995 GOTO 1100
1000 CLS
1010 PRINT:PRINT:PRINT:PRINT
1020 PRINT TAB(4); "EL PUNTO P NO ES EXTERIOR A LA"
1030 PRINT "RECTA R, NO DETERMINAN UN PLANO."
1040 GOTO 250
1100 REM *****
1105 REM * EDICION EN LA PANTALLA *
1110 REM * DE LOS RESULTADOS *
1115 REM *****
1120 CLS
1125 PRINT:PRINT
1130 PRINT "FORMA PARAMETRICA DE LA ECUACION"
1140 PRINT "DEL PLANO:"
1160 PRINT
1170 PRINT "X = ";P1;" + ";U1;"t + ";V1;"s"
1180 PRINT "Y = ";P2;" + ";U2;"t + ";V2;"s"
1190 PRINT "Z = ";P3;" + ";U3;"t + ";V3;"s"
1200 PRINT:PRINT
1210 PRINT "FORMA GENERAL DE LA ECUACION"
1220 PRINT "DEL PLANO:"
1230 PRINT
1240 LET A=U2*V3-V2*U3
1250 LET B=U3*V2-U2*V3
1260 LET C=U1*V2-V1*U2
1270 LET D=-(A*P1+B*P2+C*P3)

```

APRENDER CON EL ORDENADOR

```
1280 PRINT
1285 PRINT A;"x" + "B;"y" + "C;"z" + "D;" = "Q"
1286 FOR I=1 TO 1000:NEXT I:PRINT :PRINT :PRINT
1288 PRINT :PRINT :PRINT :PRINT "(ESPERA UN MOMENTO, POR FAVOR)"
1290 FOR I=1 TO 10000:NEXT I:CLS:GOTO 250
```

Programa 2. Espacio afín. Ecuaciones del planoPLANO1.BAS

Modificaciones para otros equipos

Son necesarias las siguientes modificaciones para otros equipos:

ZX-SPECTRUM:

```
95 CLS
220 IF A=1 THEN GOTO 320
222 IF A=2 THEN GOTO 500
224 IF A=3 THEN GOTO 800
226 GOTO 95
310 FOR I=1 TO 2000: NEXT I: GOTO 9999
```

COMMODORE:

```
45 PRINT CHR$(147)
240 PRINT CHR$(147)
280 IF A$ <> "N" AND A$ <> "n" THEN
PRINT CHR$(147): GOTO 260
285 PRINT CHR$(147)
345 PRINT CHR$(147)
450 PRINT CHR$(147)
460 IF U1*V2=V1*U2 AND U2*V3=V2*U3
THEN PRINT "LOS VECTORES
<U> Y <V> SON PROPORCIONALES, NO DETERMI-
NAN UN PLANO.":
FOR I=1 TO 3000: NEXT I: PRINT CHR$(147):
GOTO 250
510 PRINT CHR$(147)
628 PRINT CHR$(147)
700 IF U1*V2=U2*V1 AND U2*V3=V2*U3 THEN
PRINT "LOS PUNTOS P, Q, Y R ESTÁN ALINEADOS, NO
DETERMINAN UN PLANO.": FOR I=1 TO 3000: NEXT I:
PRINT CHR$(147): GOTO 250
810 PRINT CHR$(147)
1000 PRINT CHR$(147)
1120 PRINT CHR$(147)
1290 FOR I=1 TO 10000: NEXT I: PRINT
CHR$(147):GOTO 250
```

SOCIEDAD

■ La pintura española en el siglo XX

La renovación de la pintura española contemporánea fue iniciada ya a finales del siglo XIX y principios del XX por Beruete, Sorolla y Zuloaga, principalmente. Sin embargo, fue en Barcelona donde a raíz de 1890 comenzó el gran movimiento de renovación pictórica. El principal de estos pintores renovadores

barceloneses es Ramón Casas, uno de los mejores impresionistas españoles y extraordinario dibujante. A su misma generación pertenece Santiago Rusiñol, y a la siguiente Isidro Nonell. Coetáneo de éstos es José María Sert, extraordinario muralista.

Poco después de haberse iniciado el movimiento renovador barcelonés, comenzó en París el movimiento cubista, donde destacaron los españoles: Picasso y Juan Gris. Picasso se inició con pintura modernista y de gran colorido. Posteriormente, en 1905 se trasladó a Madrid el movimiento renovador, pues estuvo en París, donde se relacionó con Juan Gris y Picasso.

Por esta época José Gutiérrez Solana se centra en la pintura expresionista.

De Cataluña también provienen las dos máximas figuras del surrealismo español: Joan Miró y Salvador Dalí.

En Madrid, en torno a Benjamín Palencia, se formaron una serie de pintores jóvenes que constituirían en 1945 la Tercera Escuela de Madrid, que mostró gran interés por el paisaje. También de Madrid destacan: Ortega Muñoz y Rafael Zabaleta.

Al mismo tiempo surgieron diversos grupos de pintores que se incorporaron a las nuevas tendencias surgidas en el extranjero y fueron los introductores en España de la pintura abstracta. A uno de estos grupos pertenecen: Juan José Tharrats, Cuixart, Ponç y Tapiés.

A partir de 1954 triunfaba la pintura no imitativa en la Cuarta Escuela de Barcelona, formada por los pintores catalanes Cuixart, Tapiés y Tharrats.

En Madrid, la pintura no imitativa la desarrolló un grupo de pintores: Canogar, Feito, Millares y Saura, a quienes se incorporaron un año después Rivera, Viola y el escultor Martín Chirino.

Todavía en esta época de los cincuenta, especialmente rica desde el punto de vista artístico y cultural, aparecieron dos nuevos movimientos opuestos al informalismo: Arte

normativo español (de carácter analítico y experimental) y el Realismo social.

En fechas más recientes surgieron dos

nuevos movimientos pictóricos: Nueva figuración y el Realismo crítico (Genovés, Canogar, etcétera)

```
10 REM *****
15 REM * CONOCIMIENTOS SOBRE PINTURA *
20 REM *   ESPAÑOLA DEL SIGLO XX   *
25 REM *****
30 CLS
40 LET A=0
50 RANDOMIZE TIMER
60 DIM A$(10): DIM B$(40): DIM N(10)
70 REM *****
75 REM * LECTURA DE PREGUNTAS *
80 REM *****
90 FOR I=1 TO 10
100 READ A$(I)
110 NEXT I
120 REM *****
125 REM * LECTURA DE RESPUESTAS *
130 REM *****
140 FOR J=1 TO 40
150 READ B$(J)
160 NEXT J
170 REM *****
175 REM * LECTURA DE SOLUCIONES *
180 REM *   A LAS RESPUESTAS   *
185 REM *****
190 FOR K=1 TO 10
195 READ N(K)
200 NEXT K
210 REM *****
215 REM * EDICION EN PANTALLA DE *
220 REM * PREGUNTAS Y RESPUESTAS *
225 REM *****
230 FOR I=1 TO 10
240 PRINT:PRINT TAB(2);A$(I)
250 PRINT:PRINT
260 PRINT:PRINT TAB(6);B$(I*4-3)
270 PRINT:PRINT TAB(6);B$(I*4-2)
280 PRINT:PRINT TAB(6);B$(I*4-1)
290 PRINT:PRINT TAB(6);B$(I*4)
300 PRINT:PRINT
310 REM *****
315 REM * INTRODUCCION DE LA *
320 REM * SOLUCION CORRECTA *
325 REM *****
330 INPUT "LA RESPUESTA ES: ";R
340 IF R=N(I) THEN GOSUB 500
350 IF R<N(I) THEN GOSUB 800
360 NEXT I
370 PRINT:PRINT:PRINT
380 REM *****
385 REM * EDICION DEL FIN DEL *
390 REM *   PROGRAMA   *
395 REM *****
400 PRINT TAB(3);"RESPUESTAS CORRECTAS: ";A
410 PRINT:PRINT
415 IF A<1 OR A>10 THEN GOTO 400
420 IF A=1 AND A<4 THEN PRINT "MUY DEFICIENTE":GOTO 490
430 IF A=4 THEN PRINT "INSUFICIENTE":GOTO 490
440 IF A=5 THEN PRINT "SUFICIENTE":GOTO 490
450 IF A=6 THEN PRINT "BIEN":GOTO 490
460 IF A>6 AND A<9 THEN PRINT "NOTABLE":GOTO 490
470 IF A=9 THEN PRINT "SOBRESALIENTE":GOTO 490
480 PRINT "MUY BIEN. PERFECTO. TIENES UN 10 DE NOTA"
490 FOR I=1 TO 2000:NEXT I:CLS:END
500 REM *****
505 REM * SUBROUTINA DE RESPUESTAS *
510 REM *   CORRECTAS   *
515 REM *****
520 CLS
530 PRINT:PRINT
540 PRINT " ¡SOLUCION CORRECTA!"

550 LET A=A+1
560 FOR F=0 TO 2000:NEXT F:CLS
570 RETURN
```

```

800 REM *****
805 REM * SUBROUTINA DE RESPUESTAS *
810 REM *      INCORRECTAS      *
815 REM *****
820 CLS
830 PRINT:PRINT:PRINT
840 PRINT " ;NO ES CORRECTO!"
850 PRINT:PRINT
860 PRINT "LA SOLUCION CORRECTA ES: "
865 PRINT:PRINT B$(I*4+N(I)-4)
870 FOR F=1 TO 5000: NEXT F:CLS
880 RETURN
1000 REM *****
1010 REM * DATAS CON LAS PREGUNTAS *
1020 REM *   RESPUESTAS Y CON LAS *
1030 REM *   SOLUCIONES CORRECTAS *
1040 REM *****
1050 DATA "PINTOR BARCELONES GRAN IMPRESIONISTA Y RENOVADOR DE FINALES DEL XIX:"
1060 DATA "¿CUAL DE ESTAS EPOCAS NO PERTENECE A LA PINTURA DE PICASSO?"
1070 DATA "MURALISTA DESTACADO, COETANEO DE NONELL:", "MAXIMAS FIGURAS DEL SURREA-
LISMO ESPAÑOL:"
1080 DATA "¿QUE DOS MOVIMIENTOS APARECEN EN LOS CINCUENTA?", "CUIXART Y TAPIES A
QUE MOVIMIENTO PERTENECEN:"
1090 DATA "EN PARIS, PINTORES ESPAÑOLES DESTACADOS DEL CUBISMO:", "PINTOR INICIA-
DO DEL CUBISMO EN MADRID:"
1100 DATA "EN MADRID, PINTORES PERTENECIENTES A LA <PINTURA NO IMITATIVA>:", "¿QU
E DOS MOVIMIENTOS SIGUIERON TAPIES Y CUIXART"
1110 DATA "1. ZULOAGA", "2. JUAN GRIS", "3. SOROLLA", "4. RAMON CASAS", "1. AZUL Y
ROSA", "2. DINARD Y DE METAMORFOSIS", "3. CUBISTA Y CLASICA", "4. DINARD Y NORMATIV
A"
1120 DATA "1. JUAN GRIS", "2. RUSINOL", "3. JOSE MARIA SERT", "4. VAZQUEZ DIAZ", "1.
CANOGAR Y TAPIES", "2. JOAN MIRO Y DALI", "3. MIRO Y CUIXART", "4. VIOLA Y DALI"
1130 DATA "1. ARTE NORMATIVO Y PAISAJISMO", "2. ARTE NORMATIVO Y REALISMO SOCIAL"
, "3. REALISMO SOCIAL Y ABSTRACTO", "4. REALISMO CRITICO Y SURREALISMO"
1140 DATA "1. PINTURA ABSTRACTA", "2. EXPRESIONISMO", "3. SURREALISMO", "4. DADAISM
O", "1. JOAN MIRO Y PICASSO", "2. PICASSO Y TAPIES", "3. JUAN GRIS Y CANOGAR", "4. P
ICASSO Y JUAN GRIS"
1150 DATA "1. VAZQUEZ DIAZ", "2. VIOLA", "3. CUIXART", "4. PICASSO", "1. PONS, CANO
GAR Y FEITO", "2. MILLARES, FEITO Y THARRATS", "3. CANOGAR, FEITO Y SAURA", "4. GENO
VES, SAURA Y MILLARES"
1160 DATA "1. SURREALISMO Y CUBISMO", "2. REALISMO Y ABSTRACTO", "3. ABSTRACTO Y P
INTURA NO IMITATIVA", "4. CUBISMO Y PAISAJISMO"
1170 DATA 4,4,3,2,2,1,4,1,3,3

```

Programa 3. Conocimientos sobre pintura española del siglo XXTESTPIN2.BAS.

Modificaciones para otros equipos

Son necesarias las modificaciones si-
guientes:

ZX-SPECTRUM:

```

50 (NO ES NECESARIA)
60 DIM A$(10,40): DIM B$(40,30): DIM N(10)
490 FOR I=1 TO 1000: NEXT I: CLS: GOTO 9999
500 FOR F=1 TO 1000: NEXT I: CLS
870 FOR F=1 TO 1000: NEXT I: CLS

```

COMMODORE:

```

30 PRINT CHR$(147)
490 FOR I=1 TO 2000: NEXT I: CHR$(147): END
500 FOR I=1 TO 2000: NEXT I: CHR$(147)
870 FOR F=1 TO 5000: NEXT I: CHR$(147)

```

PARA LOS MAS JOVENES

■ Expresiones algebraicas. Suma y diferencia de polinomios

Una *expresión algebraica* es un conjunto de letras o de números y letras ligados por los signos de las operaciones algebraicas.

Son ejemplo de expresiones algebraicas:

- $x + 1 = 3$
- $x^2 + 2x - 3$
- $x - 2x^4$

Cada letra significa un número "no terminado".

1. Monomios y polinomios

La expresión algebraica más sencilla se denomina **monomio** y lo constituyen números y letras ligados por el signo de multiplicar.

Como ejemplo tienes:

$3x^2$; x^5 ; $7/x$; $-x$.

El número que multiplica al conjunto de letras se llama *coeficiente*, y las letras se denominan *parte literal*. El grado de un monomio es la suma de exponentes de sus letras.

Un **polinomio** es una suma de monomios. Cada monomio es un término del polinomio. El grado del polinomio es el del monomio, que lo tiene mayor.

1.1. Suma y diferencia de polinomios

Suma. Para sumar varios polinomios se

escriben unos a continuación de otros formando un solo polinomio con ellos y se reducen a términos semejantes (aquéllos con la misma parte literal).

1.2. Diferencia de polinomios

Para restar polinomios se suma al minuendo el sustraendo cambiado de signo.

Este *programa* suma y resta polinomios. Primeramente tienes que introducirle los grados de cada uno de los polinomios y, posteriormente, él te irá pidiendo los coeficientes de cada uno de los monomios que los componen. Al final, en la pantalla te muestra los resultados de la suma y de la resta, pero como coeficientes, de tal modo que tú luego los pondrás delante de cada parte literal correspondiente.

```
10 REM *****
12 REM *
15 REM * SUMA Y DIFERENCIA DE *
17 REM *
20 REM * POLINOMIOS *
22 REM *
25 REM *****
30 CLS
40 PRINT :PRINT
50 PRINT TAB(4);"SUMA Y RESTA DE POLINOMIOS"
60 PRINT TAB(4);"-----"
70 PRINT :PRINT
80 PRINT TAB(3);"PARA PODER SUMAR O RESTAR DOS PO"
90 PRINT "LINOMIOS TIENES QUE:"
100 PRINT : FOR I=1 TO 500:NEXT I
110 PRINT "-- INTRODUCIR EL GRADO DE DOS"
120 PRINT TAB(3);"POLINOMIOS A OPERAR."
130 PRINT
135 FOR I=1 TO 500:NEXT I
140 PRINT "-- INTRODUCIR LOS COEFICIENTES DE"
150 PRINT TAB(3);"LAS <X> DE MAYOR A MENOR GRADO"
160 PRINT TAB(3);"DE CADA UNO DE ELLOS."
170 PRINT :PRINT :PRINT :PRINT
175 FOR I=1 TO 3500:NEXT I
180 INPUT "(PULSA UNA TECLA PARA CONTINUAR)";P$
190 CLS
200 PRINT :PRINT
210 REM *****
212 REM * INTRODUCCION GRADOS DE *
214 REM * DE LOS POLINOMIOS *
215 REM *****
217 PRINT :PRINT :PRINT :PRINT
220 INPUT "GRADO POLINOMIO-1 :";N
225 PRINT :PRINT :PRINT
230 PRINT :PRINT
240 INPUT "GRADO POLINOMIO-2 :";M
245 CLS
250 IF N>M THEN LET P=N: GOTO 270
260 LET P=M
270 DIM A(P),B(P),S(P),R(P)
280 REM *****
282 REM * PONER COEFICIENTES *
284 REM * A CERO *
286 REM *****
290 FOR I=0 TO P
300 LET A(I)=0
310 LET B(I)=0
320 LET S(I)=0
330 LET R(I)=0
340 NEXT I
350 PRINT :PRINT
360 PRINT "COEFICIENTES DEL POLINOMIO-1:"
```

```

370 PRINT
380 FOR I=N TO 1 STEP -1
390 PRINT "COEFICIENTE DE X^";I;":":INPUT A(I)
400 NEXT I
410 INPUT "TERMINO INDEPENDIENTE:":A(0)
420 CLS
430 PRINT :PRINT
440 PRINT "COEFICIENTES DEL POLINOMIO-2:"
460 FOR I=M TO 1 STEP -1
470 PRINT "COEFICIENTE DE X^";I;":":INPUT B(I)
480 NEXT I
490 INPUT "TERMINO INDEPENDIENTE:":B(0)
500 CLS
501 REM *****
502 REM * SUMA DE POLINOMIOS *
503 REM *****
505 PRINT :PRINT
510 PRINT TAB(7);"SUMA DE POLINOMIOS"
520 PRINT TAB(7);"-----"
530 PRINT :PRINT TAB(3);"EL POLINOMIO SUMA ES:"
540 PRINT
550 FOR I=P TO 1 STEP -1
560 S(I)=A(I)+B(I)
570 PRINT "COEFICIENTES DE X^";I;":":S(I)
580 NEXT I
590 PRINT "TERMINO INDEPENDIENTE:":A(0)+B(0)
600 PRINT:PRINT:PRINT
700 FOR I=1 TO 1000:NEXT I:INPUT "(PULSA UNA TECLA PARA CONTINUAR)":P$
710 CLS
720 REM *****
722 REM * RESTA DE POLINOMIOS *
725 REM *****
730 PRINT :PRINT
734 PRINT TAB(7);"RESTA DE POLINOMIOS"
736 PRINT TAB(7);"-----"
740 PRINT TAB(3);"EL POLINOMIO DIFERENCIA ES:"
750 PRINT
760 FOR I=P TO 1 STEP -1
770 LET R(I)=A(I)-B(I)
780 PRINT "COEFICIENTE DE X^";I;":":R(I)
790 NEXT I
800 PRINT "TERMINO INDEPENDIENTE:":A(0)-B(0)
810 PRINT :PRINT :PRINT
820 INPUT "(PULSA UNA TECLA PARA CONTINUAR)":P$
825 CLS
830 PRINT :PRINT :PRINT :PRINT
840 PRINT:PRINT "DESEAS OPERAR CON OTROS DOS POLI"
850 PRINT "NOMIOS (S/N)";
860 INPUT K$
870 IF K$="S" OR K$="s" THEN CLS:GOTO 40
880 IF K$<>"N" AND K$<>"n" THEN CLS:GOTO 830
890 CLS
900 LOCATE 10,12:PRINT "   ¡¡¡ ADIOS !!! "
910 FOR I=1 TO 3000:NEXT I:CLS
920 END
    
```

Programa 4. Suma y diferencia de polinomiosPOLINOM1.BAS.

Modificaciones para otros equipos

ZX-SPECTRUM:

```

900 PRINT AT 10,12:PRINT "¡¡¡ADIOS!!!"
920 GOTO 9999
    
```

COMMODORE:

```

30 PRINT CHR$(147)
190 PRINT CHR$(147)
245 PRINT CHR$(147)
420 PRINT CHR$(147)
500 PRINT CHR$(147)
    
```

```

710 PRINT CHR$(147)
825 PRINT CHR$(147)
870 IF K$="S" OR K$="s" THEN PRINT
CHR$(147):GOTO 40
880 IF K$<>"N" AND K$<>"n" THEN PRINT
CHR$(147):GOTO 830
890 PRINT CHR$(147)
900 PRINT "¡¡¡ADIOS!!!"
910 FOR I=1 TO 3000: NEXT I: PRINT CHR$(147)
    
```

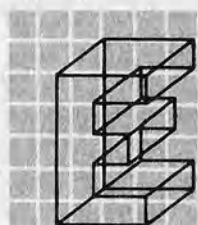
AMSTRAD:

```

900 LOCATE 10,12: PRINT "¡¡¡ADIOS!!!"
    
```

PEQUEÑA HISTORIA DE LA INFORMATICA

■ El ENIAC: ¿calculador u ordenador?



N 1935 la Armada de los Estados Unidos de América había establecido un laboratorio para investigación sobre balística, en Aberdeen, una pequeña localidad de Maryland. El objetivo funda-

mental a cumplir por este laboratorio era la elaboración de unas tablas de balística, elemento considerado indispensable para el Ejército. El cálculo de dichas tablas no era especialmente complicado, aunque sí requería multitud de operaciones. Una tabla útil requería calcular muchísimas trayectorias (de 2.000 a 4.000), y para cada trayectoria era necesario efectuar una media de 750 multiplicaciones. Como todas estas operaciones suponían un tiempo y esfuerzo enormes, el laboratorio de investigación consideró de importancia fundamental encontrar algún medio de acelerar estos cálculos exhaustivos. Naturalmente, el laboratorio contaba con un consejo científico consultivo, en el que se encontraban ilustres matemáticos y físicos, entre ellos John von Neumann (que será protagonista en otro momento de nuestra historia), y Hermann Goldstine, que estaba entonces haciendo su servicio militar (realmente estaba movilizado).

Desde el año 1930, el Ballistic Research Laboratory contaba con colaboradores, en Universidades privadas y estatales, y entre ellas era importante la aportación de la Moore School of Electrical Engineering, de la Universidad de Pensilvania, en Filadelfia. Dicha Universidad era muy activa, y en el año 1941 organizó un curso de verano para el estudio de la electrónica, invitando a personalidades relacionadas con esa materia. Entre los invitados se encontraba el director del Departamento

de Física del Ursinus College, en Pasadena, que estaba encargado de uno de los cursos impartidos. El profesor Mauchly estaba muy interesado en los calculadores, y había estado trabajando en intentar mejorar las velocidades de cálculo de estas máquinas. Como organizador del curso, estaba J. Presper Eckert, muy joven y estudioso de los tubos de vacío, y en general de la electrónica, interesándose mucho menos por los problemas matemáticos de la Física. Que estos dos hombres que residían en dos ciudades distintas se conocieran fue una casualidad que haría avanzar la ciencia informática un paso enorme. Probablemente si no hubieran sido estos dos hombres los creadores del ENIAC, otros hubieran creado un calculador semejante, ya que el terreno estaba abonado en esos años para ese tipo de trabajos, pero es indudable que fue una suerte que se conocieran y colaboraran, ya que eran dos personas complementarias.

Después de las jornadas veraniegas en la Moore School (y a pesar del clima de veranos calientes y húmedos de Filadelfia), Mauchly quedó encantado con los trabajos realizados en la Moore School y pasó a esta Universidad en septiembre del mismo año. Se puso a trabajar en el diseño de mejoras para aumentar la velocidad de los calculadores aprovechando las cualidades de los tubos de vacío, tan estudiados por su amigo, el profesor Eckert. Pocos meses más tarde, Hermann Goldstine fue también contratado por dicha Universidad, interesándose inmediatamente por los trabajos de Mauchly y Eckert.

En 1943 (con Goldstine trabajando en la Moore School), dicha Universidad obtuvo un contrato con la Ballistic Research Laboratory para el diseño y construcción de un calculador electrónico. El trabajo comenzó inmediatamente. Una de las primeras decisiones a tomar fue la de sustituir o no los relés por tubos

PEQUEÑA HISTORIA DE LA INFORMATICA

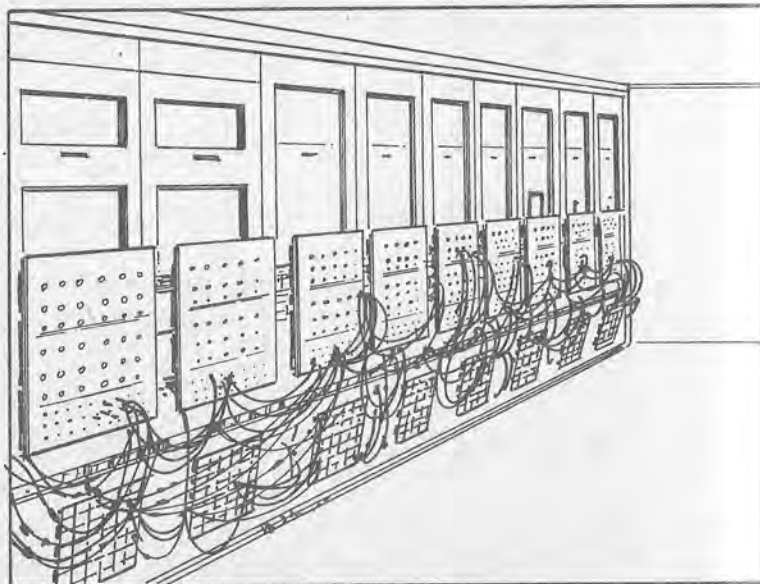
de vacío. Por un lado, los relés eran muy fiables, por otro, las velocidades de cálculo que podían obtenerse con los tubos de vacío eran muy superiores, ya que se trataba simplemente de mover electrones, no piezas mecánicas. Sin embargo y a pesar del gran amor del profesor Eckert por los tubos de vacío, éstos no eran en absoluto "formales", y se estropeaban con una facilidad pasmosa, creando enormes complicaciones, ya que inutilizaban los resultados, obligando a recomenzar las operaciones. Si además, se considera que la máquina debía llevar 18.000 tubos, es fácil imaginar los problemas que surgirían durante su funcionamiento. Realmente, que el calculador funcionara era un milagro (así lo comentaba Goldstine), ya que además de los tubos conflictivos contaba con 70.000 resistencias, 6.000 conmutadores y 10.000 condensadores.

El profesor Eckert seguía probando y considerando distintos tubos de vacío, intentando disponer de los más fiables. Seleccionó unos desarrollados en un laboratorio de física nuclear, que ofrecían mejores características que los normales.

Y, finalmente, el equipo diseñó y construyó un enorme calculador electrónico universal, que pasó a Ballistic Research Laboratory, de Aberdeen, en 1946, trabajando varios años en la Armada a entera satisfacción. La presentación de la máquina, en febrero de 1946, constituyó todo un éxito, funcionando correctamente durante toda la demostración. (Es importante notar que los programas de la máquina habían sido creados por otra programadora, la esposa de Hermann Goldstine, Adele).

Este enorme calculador u ordenador fue, en realidad, la máquina que marcó la transición entre los calculadores y los ordenadores. Fue el resultado de la demanda del Ejército de los Estados Unidos, que necesitaba este tipo de calculadores y de un país preparado para aceptarles, donde los cerebros más importantes estaban interesados en desarrollarles y potenciarles, tanto para su propio uso, como para las instituciones que se los encargaban.

En esos años se habían estado construyendo calculadores analógicos, electromecánicos y electrónicos, simultáneamente, no unos tras otros. Muchos investigadores en este campo no creyeron en la electrónica hasta años después. Otros, desde un primer momento, se dedicaron a mejorar los componentes electrónicos para ir dotando de más fiabilidad y operatividad a las nuevas máquinas calculadoras.



El ordenador electrónico ENIAC, instalado en la Moore School of Engineering, Universidad de Pensilvania, en 1946.

El ENIAC, a pesar de ser un calculador muy útil, y disponer de un diseño mucho más moderno que el MARK I, fue mucho menos conocido. Ello se debió en gran medida a que la Armada no deseaba propaganda de la máquina, que fue clasificada como "secreto militar", ya que, por aquel entonces, el público se seguía extasiando con el MARK I, el primer calculador realmente popular.

El diseño de la máquina supuso el trabajo de un año para el equipo. Su construcción duró algo más: casi dos años. Su coste era prohibitivo, sólo un organismo estatal podría pagarlo: 500.000 dólares. Goldstine, que había pertenecido al consejo científico del Ballistic Research Laboratory, sería el encargado de ocuparse de las relaciones con dicho organismo.

Ya hemos hablado de Atanasoff y su máquina ABC. Desde luego, sí es cierto que nuestros diseñadores del ENIAC visitaron al profesor de Iowa, que les mostró sus trabajos sin sospechas de ningún tipo. También es cierto que Eckert y Mauchly estaban bastante mejor apoyados que Atanasoff. Cuando les fue encargado el proyecto, el Ballistic Research Laboratory utilizaba a cien mujeres para realizar los cálculos (además de emplear pequeñas calculadoras). Los resultados, además, no eran demasiado satisfactorios. Mauchly, verdadero motor del proyecto, escribió un informe para el Ballistic Research Laboratory, pero

este informe se perdió lamentablemente. En 1943 Eckert le ayudó a reconstruirlo, tomando las notas taquigráficas de una secretaria. Lo necesitaban imprescindiblemente para una reunión, a la que asistiría el coronel Leslie E. Simon, director del laboratorio de Aberdeen, y otras personalidades, como el director del Instituto de Estudios Avanzados, de Princeton, Oswald Veblen, un insigne matemático. Veblen quedó absolutamente convencido con la exposición del proyecto y aconsejó a Simon que lo apoyara. El resultado, el ENIAC, dejó obsoleta a otra máquina, el Mark I, que había sido construida muy pocos años antes (prácticamente, estas dos máquinas son de la misma época, aunque el ENIAC fuera 5.000 veces más rápido que el Mark I, mucho más popular que él).

Mauchly y Eckert fueron hábiles en todo (o casi todo). Su máquina no superó el presupuesto, que aunque era muy alto, era fácilmente alcanzable, dado el tipo de máquina que se pretendía crear. Tampoco sobrepasaron el tiempo calculado en el proyecto para su fabricación, dejando posibles mejoras y modificaciones para otra ocasión en que pudieran efectuarlas. Además, tuvieron suerte, ya que el "ordenador", a pesar de sus tremendas complicaciones, resultó ser bastante fiable (desde luego mucho más de lo esperado).

Como sabemos, el ENIAC estaba en parte basado en los trabajos de Atanasoff. Sin embargo, la máquina del profesor de Iowa utilizaba sistema binario, y el ENIAC sistema decimal (siendo mucho más cómodo el binario). Además, tampoco estableció una clara separación entre la unidad aritmética y la memoria, como ocurría en el ABC, característica ésta también útil. Pero el ENIAC era programable (lo que no ocurría con la máquina de Atanasoff) y muchísimo más rápida en los cálculos.

La programación del ENIAC era toda una tarea. Para empezar, era necesario colocar manualmente en la posición adecuada miles de conmutadores, y además ir introduciendo también muchos cables y puentes en los puntos adecuados. En cuanto a los delicados tubos de vacío, la máquina contaba con casi 18.000. El primer mes se estropearon cincuenta (para anular los correspondientes cálculos bastaba con que fallase uno). Sin embargo, en los meses siguientes, la media bajó muchísimo, estropeándose sólo quince al mes. De todas formas, al finalizar el primer año de funcionamiento, prácticamente la totalidad de los

tubos habían sido sustituidos entre comprobaciones y sustituciones.

Veamos ahora algunas de las características técnicas de la máquina: ésta contaba con una unidad de proceso, memoria y entrada y salida de datos. Utilizaba sistema decimal y era unas cinco mil veces más veloz que el Mark I. Estaba compuesta por unos cuarenta enormes paneles que pesaban unas treinta toneladas, y ocupaba una superficie de unos 160 metros cuadrados, en dos plantas de la Moore School. Tenía 6.000 conmutadores manuales, 18.000 tubos de vacío, 70.000 resistencias, 1.500 relés y 10.000 condensadores. El motor utilizado para su accionamiento ofrecía una potencia similar a dos coches. El sistema de refrigeración también era muy complejo, dado el número de resistencias y tubos era equivalente a unas cincuenta calefacciones domésticas). Existe la anécdota de que cuando se conectaba la máquina (que consumía 150.000 vatios) las luces del barrio oeste de Filadelfia se apagaban. Naturalmente, esta pequeña anécdota no parece cierta, pero desde luego sí tiene sentido. Lo que es cierto, es que la máquina sólo podía trabajar durante una hora, teniéndose que apagar a continuación, para dejar actuar a la refrigeración. De todos modos, el ENIAC realizaba en una hora lo que el Mark I realizaba en una semana.

Otra pequeña anécdota, más creíble que la anterior, nos habla de una avería producida una noche en el ENIAC. Tras duros trabajos de localización, se encontró una pequeña polilla, que producía un cortocircuito. Los americanos utilizan desde entonces en muchos casos el término "desinsectación" frente al de "depuración", más acorde con las máquinas actuales.

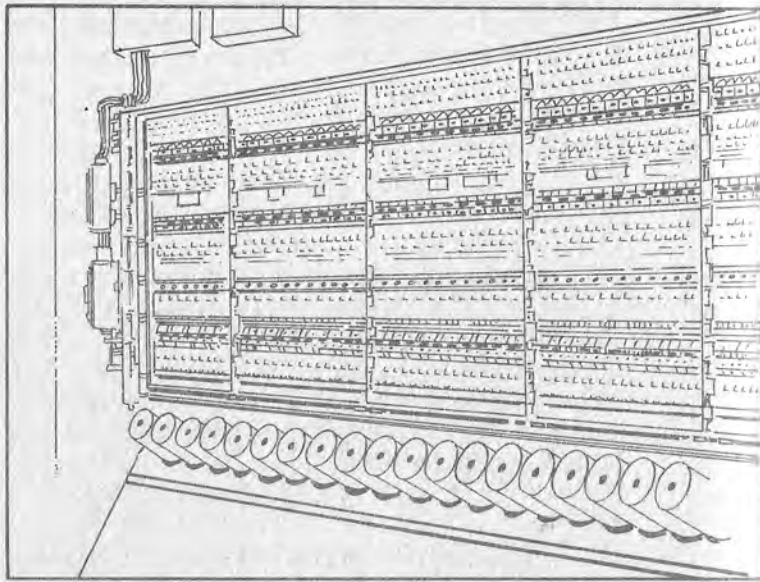
A pesar de las averías, la técnica que utilizaba válvulas electrónicas triunfó plenamente frente al sistema de relés. En el ENIAC, un dígito decimal se representaba en una forma que puede considerarse como el equivalente electrónico de una rueda de contador. A este circuito se le denomina "contador anular", y tiene dos posiciones estables diferentes, en cada una de las cuales sólo una de las diez válvulas similares lleva corriente. Cuando se recibe un pulso por el circuito, la válvula encendida se apaga y se enciende la cero, produciéndose una señal para indicar que debe llevarse a cabo un acarreo, de la misma forma que cuando una rueda de contador pasa de la posición 9 a la 0. En cuanto a memoria, el ENIAC disponía de veinte acumuladores, cada

PEQUEÑA HISTORIA DE LA INFORMATICA

uno de los cuales constaba de diez contadores anulares, uno para cada dígito del sistema decimal, contando además con otro dispositivo adicional que almacenaba el signo del número. También disponía de dispositivos de acarreo muy completos para que el número almacenado en un acumulador se pudiera sumar directamente a otro almacenado en otro acumulador. Los números en tránsito por el ENIAC estaban representados por series de pulsos eléctricos que circulaban por un conjunto de once cables, uno para cada dígito, más el del signo. El acumulador recibía el estímulo para emitir en forma de pulsos en número que tenía almacenado, de forma que cada contador anular emitía por separado un número de pulsos igual al dígito decimal contenido en él. El flujo continuo de pulsos era suministrado por un generador cuya velocidad era de 100.000 pulsos por segundo. La máquina disponía de dos juegos de conexiones independientes: una de ellas controlaba los pulsos digitales (que representaban los números), y la otra los pulsos de control de activación de las diversas unidades de la máquina, para que actuaran en la secuencia debida. Con todo lo expuesto puede deducirse que la preparación o programación de la máquina para los cálculos era extremadamente larga, y exigía mucho orden y cuidado. Sin embargo, una vez realizada, la máquina ya quedaba programada y no ofrecía mayores problemas al respecto. En cualquier caso, piénsese en las dificultades de cambio de unos cálculos a otros, para los que se requería "reprogramar" la máquina.

Con sus veinte acumuladores, la memoria del ENIAC era muy pequeña. Es probable que la pequeña memoria y la complicada programación fueran los dos problemas principales de la máquina, que, por lo demás, resultó aceptablemente fiable, y rapidísima en los cálculos, hasta el punto de poder decir que esa revolucionara por su velocidad de cálculo.

Sin embargo, Eckert y Mauchli conocían perfectamente los límites de su máquina. A pe-

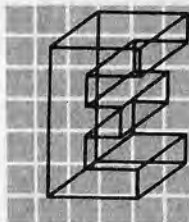


El ordenador del Institute for Advance Studies.

sar de su potencia y velocidad, y de haber seleccionado para ésta una tecnología de vanguardia, su organización estaba basada en los mismos principios de cálculo que las máquinas de Pascal y Shickard. Estos dos grandes hombres no se contentaron con lo obtenido, pero de esto ya hablaremos más adelante.

En 1946, en la Moore School de Filadelfia, además de la puesta a punto del ENIAC, tuvo lugar otro acontecimiento de importancia para la Historia de la informática, el primer curso de conferencias sobre la "Teoría y Técnicas de los Computadores Electrónicos y Digitales", cuyos planteamientos han coincidido con los seguidos en los ordenadores electrónicos futuros. Sin embargo, no podemos ni considerar que aquellos sabios señores reunidos en la Conferencia llegaran a imaginar que ordenadores pequeñísimos y baratísimos pudieran llegar a superar la potencia y características técnicas del enorme mastodonte orgullo de la Moore School, y el resto de la humanidad "civilizada" de aquellas fechas.

DINERO ELECTRONICO Y TELEBUROTICA



El sector bancario (naturalmente provisto de unos fondos que a veces faltan en otros sectores) ha sido uno de los primeros sectores en interesarse por la telemática, para ofrecer distintos servicios a sus clientes y

para agilizar sus propios procesos internos. Este interés ha ido aumentando desde la década de los setenta, y quizás una de sus miradas más importantes ha sido ir descentralizando en lo posible todos los servicios.

En nuestros días, pese al alto coste de las máquinas, éstas terminan amortizándose, en pocos años, o, quizá, se quedan anticuadas rápidamente, y pasan a otros servicios de menor interés. Sin embargo, los salarios van aumentando de día en día, y suponen el coste mayor de cualquier organización. Por esta razón, los bancos también se han interesado en ir disminuyendo en lo posible el personal de sus agencias, e ir equipando a los clientes con terminales que unidos al ordenador central y a sus ficheros generales permiten a éstos efectuar por vía telemática las operaciones más usuales. En ocasiones, la operación seleccionada se realiza directamente, pero en otros muchos casos se podrán efectuar operaciones en diferido (estas operaciones se registran localmente en la agencia correspondiente, y al final de la jornada son enviadas al ordenador central). La transmisión no presenta problemas (desde el punto de vista electrónico), pero necesita algún tipo de soporte escrito (cheque, transferencia, título de depósito, etcétera). En la actualidad existen aplicaciones totalmente electrónicas que autorizan transacciones financieras de tipo bancario, e incluso otras en un sentido comercial más amplio.

Vamos a pasar a describir alguno de los sistemas que actualmente se utilizan para realizar este tipo de transacciones.

■ Las tarjetas provistas de pista magnética

El sistema utilizado por los bancos es alquilar a organismos privados sus redes de telecomunicación, y en general sus equipos. Estas redes tienen como misión conectar los correspondientes terminales bancarios con el ordenador central del banco de que se trate. El sistema de acceso a la red utiliza unas tarjetas provistas de una pista magnética que son enviadas a los clientes, junto con una carta en la que se les indica un código secreto asociado a esa tarjeta. En la pista magnética están inscritos los parámetros de localización y el código del cliente. Cuando éste introduce la tarjeta en el terminal, lo primero que se efectúa es la lectura de la información grabada en la pista. El cliente es conminado a continuación a introducir su código de acceso, que es controlado por el propio terminal, permitiéndole (si la identificación es positiva) proseguir con la transacción. El cliente continúa su operación, pero a partir de ese momento los datos son enviados al ordenador central, que es el que realiza las comprobaciones y autoriza (u ofrece otras posibilidades) al cliente. Este tipo de operaciones (con el consiguiente movimiento de fondos) se pueden llevar a cabo de dos modos distintos: bien en tiempo real, o bien registrándolos para su posterior realización en diferido.

Entre los terminales que realizan operaciones bancarias, los más utilizados son de dos tipos distintos:

Terminales bancarios instalados en las propias agencias donde el cliente puede realizar casi todas las operaciones normales, consultar la cuenta, depósito de cantidades o retirada de efectivo, etc.

Terminales en puntos de venta. Son terminales que instala el banco de que se trate en ciertos comercios (que pueden ser de todo tipo, desde tiendas de modas, gasolineras, etcétera). Este tipo de terminales permite que

el cliente realice la compra del artículo que vende dicha tienda, comprobando que puede hacerlo. También en este caso existen dos posibilidades para la operación: puede hacerse en tiempo real, o grabarse en la propia tienda, transmitiéndose al banco los datos de las operaciones realizadas al final del día, o cuando se haya establecido. El cliente recibe en este caso un pequeño documento creditativo de la operación realizada.

Una innovación no muy utilizada en España nos llega del país vecino, Francia. Se trata de una tarjeta de pista magnética que va provista además de un microprocesador y una pequeña memoria. El nombre de esta tarjeta «viva» es Innovatron. Las posibilidades de esta nueva modalidad de tarjeta son enormes, y están basadas principalmente en la seguridad que confiere, lo que la hace especialmente útil para transacciones confidenciales, como, por ejemplo, para documentación médica, controles de identidad para el acceso a lugares controlados, y, naturalmente, como sistema de pago muy seguro. Con esta tarjeta el cliente puede seleccionar él mismo su código de control secreto, y además añadirle otros códigos o sistemas adicionales. También puede codificar las propias operaciones, de forma que no sean legibles por cualquiera, e incluso puede conservar en la memoria los datos de las operaciones efectuadas que en su caso le servirán como comprobantes. Esta tarjeta es prácticamente inviolable (por el momento, no olvidemos que todo es alcanzable en el tiempo).

En operaciones financieras, esta tarjeta es, como antes indicamos, muy útil, sobre todo debido a su pequeña memoria. En ella puede estar inscrito, por ejemplo, la disponibilidad de cada cliente, o su crédito. Al utilizarla en cada operación, se realizarán las correspondientes cuentas, quedando anotada en la memoria la nueva situación. Cuando el cliente queda a cero, puede regular su situación y la de la tarjeta en el propio banco, quedando ya preparada para seguir llevando las cuentas y saldo del cliente de forma instantánea. Naturalmente, con la utilización de este tipo de tarjetas no son necesarias las comprobaciones realizadas por el terminal, ni consultar con el ordenador central. Desde 1980, este sistema se está introduciendo paulatinamente en Francia, donde fue probado en varias ciudades en sus comienzos.

Además de las ventajas anteriores, con

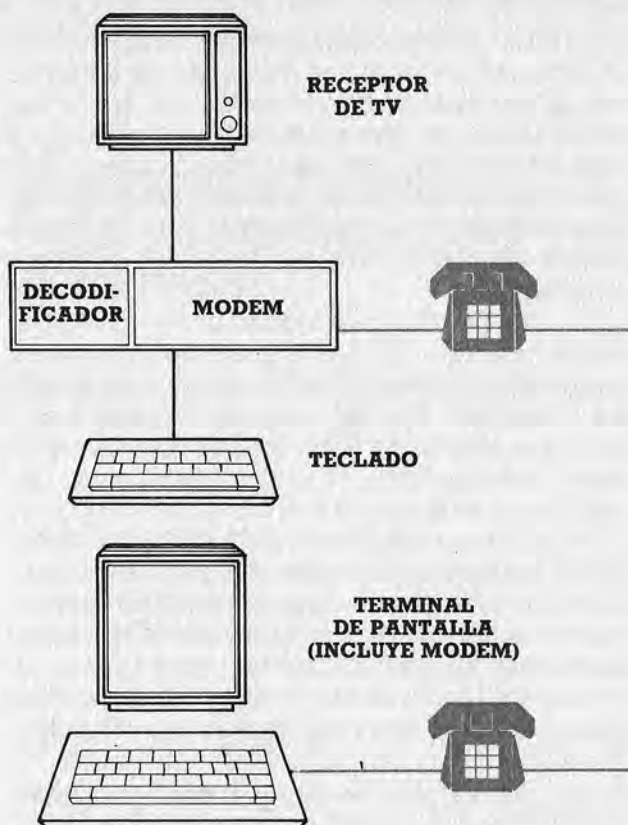
este tipo de tarjetas se podrán efectuar «telepagos» o pagos a distancia, y en general todo tipo de pagos efectuados por banco. También se podrán vender y comprar artículos por correspondencia con toda comodidad, e incluso servicios (reservas de hoteles, billetes de tren, avión, viajes de negocios o vacaciones, etcétera). Como antes indicamos, el cliente tendrá un saldo disponible y un crédito. Cuando uno de ellos se encuentre agotado, el cliente podrá acudir al banco para obtener más crédito o regularizar su saldo.

Contando con todas las ventajas anteriores, sin embargo, nosotros le encontramos muchas más posibilidades para otros usos de tipo general, e incluso para la Administración. La Seguridad Social podría llevar un control mucho más exacto de los medicamentos expedidos a cada usuario o enfermo. También podría llevar un control confidencial del expediente médico del enfermo.

Para las bibliotecas puede también ser muy útil, y para todos los organismos públicos, como el Servicio de Correos, permitiendo pagar los envíos en ciertos buzones especiales, sin necesidad de compra de sellos. En general, se pueden realizar muchísimas operaciones sin el inconveniente de tenerse que desplazar para realizarlas.

Las organizaciones de crédito privadas, como el Diners Club o American Express, tienen en esta tarjeta un aliado utilísimo. En la propia tarjeta puede ir grabado no sólo el crédito, sino además, el Organismo al que debe ir cargado ese crédito, de forma que transmitan los datos al Organismo de forma directa, durante la propia operación de lectura de la tarjeta, agilizándose la operación enormemente. El ordenador central con el que se ha comunicado el terminal contestará aceptando o denegando la operación.

Por el momento, este tipo de tarjetas parecen muy seguras, aunque, hasta nuestros días, todas las riquezas han podido ser vulneradas. Los complicados diseños de las pirámides para impedir su violación no han impedido a saqueadores acceder a sus tesoros. Tampoco los bancos son inviolables, ni las informaciones «reservadas» se han protegido siempre del acceso por personas capaces de eliminar los controles de seguridad, que las utilizan en su propio provecho. Los nuevos «carteristas» del futuro serán avisados expertos en codificación, sin demasiados escrúpulos.



■ La telemática doméstica

La telemática tiene otras muchas posibilidades. La digitalización de las imágenes de televisión permitirá convertir más fácilmente las imágenes entre los distintos sistemas utilizados (PAL, SECAM y NTSC). También ofrecerá la posibilidad de acceder a películas, programas, noticias, etc., almacenados, y que podrán ser visualizados a petición del televidente.

Naturalmente, lo anterior es también válido para programas de radio, que podrán ser seleccionados a la carta.

El teléfono mejorará su servicio y pasará a utilizar una cámara y pantalla local que emitirán la imagen además de la voz (a elección del usuario, naturalmente). Esta cámara tendrá otras múltiples posibilidades, como, por ejemplo, la lectura de documentos a distancia.

Otra ventaja adicional será poder encargar productos directamente. Su presentación podrá hacerse a través del televisor, observando las mercancías desde distintos ángulos, lo que permitirá realizar una elección casi tan perfecta como si nos hubiéramos acercado a la tienda correspondiente.

Pero esta no es la única posibilidad en el hogar: de hecho existen casas totalmente controladas electrónicamente, y a cuyas instalaciones de control se puede acceder desde el exterior mediante «control remoto» por línea telefónica (aunque a nivel experimental aún). La compañía Philips ha instalado alguna de estas casas, en las que se puede controlar desde el lavado de la ropa, asado de su filete, la calefacción, riego, cerrado de paso del gas, etcétera). Pero eso no es todo, también se puede controlar el paso de las personas conocidas, las alarmas, etc.

Sin embargo, aunque esto ya se puede adquirir en el mercado, existen a nivel más asequible prototipos de ordenadores que pueden controlar varios aparatos al mismo tiempo, adjudicando tiempos y momentos óptimos para su funcionamiento, en función de unas premisas previamente establecidas. (En suma, una especie de «ordenador ama-de-casa que decide si se lava la ropa o es mejor previamente regar el jardín, porque el día anterior fue muy caluroso»).

La telemática irá creciendo con más vigor del que ahora tiene, y se espera que para los años 1990-1995 las redes de telecomunicación se hayan extendido hasta los usuarios más reacios. Las clases mediante videotexto serán absolutamente comunes, pudiendo además seleccionar entre numerosas disciplinas y manualidades a las que nos es difícil acceder generalmente por falta de tiempo. De este modo, si se dispone de media hora o tres cuartos de hora, se puede seleccionar una clase de inglés o ruso, o electrónica para arreglar su propio televisor, o incluso (dentro de las manualidades) aprender a arreglar las persianas de su casa. Esperemos que este maravilloso futuro no sirva para que el hombre se aísle entre las cuatro paredes de su casa, donde dispone de todo, y, por el contrario, sea una vía de comunicación que ofrezca nuevas posibilidades de relación con los demás, aumentando las horas de ocio y disponibilidad del individuo.

Hasta ahora no hemos hablado de la telemática utilizada en una oficina de cualquier tipo. Con su aparición, la capacidad de los sistemas informáticos se ha acrecentado enormemente. Las memorias han aumentado de tamaño, la velocidad ha crecido mucho, y los sistemas multiproceso se multiplican de día en día.

También ha crecido el campo de la aplicación de las tareas informáticas: cada día va apareciendo un software específico que, por otro

lado, también va decreciendo de precio todos los años. La telemática permite acceder a los programas e información desde puntos lejanos. La distancia puede ser simplemente la habitación de al lado, o incluso se puede acceder al ordenador desde otro continente. Por orden de complejidad, citaremos tres tipos de aplicaciones comunes de la telemática:

Acceso a programas. El usuario, desde su terminal, puede acceder al gran ordenador central para la ejecución de determinados programas, o tomar la información necesaria para ejecutarlos en el terminal, o almacenarlos en la memoria de masa del ordenador central.

Transferencia de información y programas entre varios ordenadores o entre un terminal inteligente (microordenador) y el gran ordenador central (por ejemplo, tomando de éste la información necesaria y cargándolo en su memoria, para, una vez terminada la ejecución, salvar los resultados en la memoria del ordenador central, liberando, al mismo tiempo, la información utilizada.

Tareas compartidas. En los sistemas más evolucionados, se pueden unir dos o más ordenadores, de forma que cuando uno de ellos no pueda realizar todas las tareas que se

le piden, pueda descargarse momentáneamente pasando algunas de las tareas al otro (u otros) ordenadores conectados con él. Ya hemos hablado en otra ocasión de la precisión de estas conexiones, de los posibles sistemas de acreditar emisiones y recepciones y de las condiciones que deben cumplir las velocidades de los dispositivos que funcionan en tareas compartidas.

No hemos mencionado tampoco la posibilidad de que el usuario contrate el acceso a las grandes bases de datos disponibles desde su ordenador. El enorme campo de posibilidades que ofrece no tiene límites. Desde escritores, matemáticos, químicos, técnicos en general que deseen conocer cómo han sido desarrollados por otros, temas que les interesan, hasta juristas que deseen conocer leyes propias de determinados países, médicos, que accedan a revistas y otras publicaciones especializadas, etc., sin olvidar el acceso a bancos de datos de urgencias médicas, que puedan asesorar en la toma de decisiones sobre primeros auxilios en accidentes caseros. En suma, la telemática supone la eliminación de muchas barreras para la diseminación del saber humano, a la vez que ayuda a sobrellevar las tareas rutinarias y pesadas.

■ GLOSARIO DE TERMINOS UTILIZADOS EN TELEMATICA (CONCLUSION)

Control remoto. Sistema de actuación de dispositivos electrónicos realizado a distancia mediante el envío de códigos, bien por vía telefónica o por radio.

Aparato físico que realiza dicho control a distancia.

Crédito electrónico. Se llama así al que se efectúa a través de las tarjetas magnéticas que incluyen memoria, y donde aparece grabado el crédito disponible, que puede ser consultado por el terminal sin necesidad de acceder al ordenador central.

Valor del crédito disponible cuyo importe está grabado en las tarjetas magnéticas con memoria.

Dinero electrónico. Se engloban en este nombre todas las operaciones bancarias realizadas por procedimientos telemáticos: acceso a cuentas, ordenación de transferencias, obtención de dinero, etc.

Innovatron. Nombre comercial de la tarjeta magnética con memoria más extendida actualmente.

ISO. International Standart Organization, Organización Internacional para la Normalización en todos los campos de la Técnica y las Ciencias. En Informática no sólo ha definido códigos y normas de estandarización sino que ha creado comités para el establecimiento de recomendaciones a los constructores de ordenadores, desarrolladores de software, etc.

Modulación. Modificación de alguno de los caracteres básicos de una señal (portadora) para transmitir información. Según cuál sea la característica modificada en la señal de base, se puede hablar de modificación en frecuencia, en amplitud, fase, etc.

Pixel. Contracción de la expresión «picture-element». Unidad mínima de información componente de un gráfico. Cada una de las pequeñas áreas en las que puede dividirse la pantalla de un medio de representación de gráficos.

Sintetizador. Dispositivo utilizado en la generación de sonidos a partir de señales digitales. Se pueden utilizar bien para «sintetizar» música, mezclando canales, reproduciendo diferentes instrumentos, etc., o para la reproducción de «palabra hablada» (sintetizadores de voz).

Tarjeta magnética con memoria. Tarjeta de crédito desarrollada modernamente que incluye además de la banda magnética con los datos, códigos, identificaciones, etc., necesarios, un microprocesador y cierta capacidad de memoria que la cualifican como una tarjeta «viva»: es decir, capaz de realizar ciertas funciones, capaz de modificar los datos en ella grabados, en lugar de ser un mero soporte estático de datos.

Teleburótica. Actividades informáticas de oficina (burótica) extendidas más allá del recinto de una oficina mediante técnicas telemáticas.

Telemática. Contracción de la expresión «telecomunicación más informática». Conjunto de servicios obtenidos mediante técnicas mixtas: informáticas y de telecomunicación.

Télex. Red pública de comunicación de datos por línea telefónica. En ocasiones coloquialmente se llama télex a los teletipos (equipos periféricos de transmisión y/o recepción de datos preparados para su funcionamiento conectados a la red télex).

Multiplicando. Uno de los términos de un producto, junto con el multiplicador.

Multiproceso. Proceso por el cual se ejecutan "simultáneamente" dos o más programas o secuencias de instrucciones de ordenador por un ordenador único o una red. Normalmente el "multiproceso" no es tal, pues el ordenador en cada momento sólo realiza una tarea, aunque va intercalando pequeñas actividades de varios procesos y "parece" que los está procesando simultáneamente. Para que se produzca multiproceso deben disponerse varios procesadores, como en los ordenadores de proceso paralelo.

Multiprocesador. Ordenador que puede ejecutar uno o más programas empleando dos o más unidades de proceso bajo el control de unos dispositivos o programas (software o hardware).

Multiprogramación. Ejecución de dos o más programas que se encuentran en su memoria de forma aparentemente simultánea, por un ordenador. Los sistemas operativos modernos de alto nivel suelen disponer de esta característica.

Multivibrador biestable. Circuito electrónico fundamental en dispositivos digitales. Sinónimo de "flip-flop". Este circuito sólo puede encontrarse en dos estados estables, por lo que resulta muy útil para representar un bit de información.

Multidifusión. Sistema de transmisión de un mensaje o comando en una red o entorno complejo de ordenadores mediante el cual el envío se realiza de un modo indiscriminado a todos los posibles receptores.

Múltiple, asignación. Forma de instrucción de asignación en la que varias variables tienen asignado el mismo valor.

Multinivel, sistema de seguridad. Forma de operación en la que existen distintos niveles de seguridad para usuario y procesos. Suelen existir varios niveles de acceso en función de diferentes claves y/o puntos de entrada al sistema.

Multienlazado. Que posee un enlace con varias estructuras de datos distintas. A veces se denomina también cadena múltiple.

Multiestable. Circuito electrónico que posee varios estados estables.

Multitarea. Ejecución concurrente de varios procesos. Véase **Proceso paralelo**.

Mutua, exclusión. Relación existente entre dos o más procesos, de forma que nunca se ejecute uno de ellos mientras se está ejecutando una parte específica del otro. Es, por tanto, una relación excluyente. En ciertos sistemas operativos, en la asignación de recursos no compatibles es fundamental asegurarse de que sólo se está utilizando un periférico por parte de uno de los procesos en cada momento. Para lograrlo, se utilizan indicadores al comienzo de los puntos críticos del proceso.

Multivalía, mezcla. Operación por la cual tres o más listas se intercalan formando una única lista.

